Université de La Manouba
Ecole Nationale des Sciences de l'Informatique
Laboratoire CRISTAL - Pôle RIM

Mémoire de Mastère en Informatique

Présenté en vue de l'obtention du titre
Diplôme de Mastère en Informatique

par
Ahlem ELHAJ

Sujet :
PROPOSITION D'UN SUPPORT MULTICAST POUR LES RESEAUX MOBILES

Sous la direction de :
Pr. Abdelfettah BELGHITH (*ENSI*)
Pr. Bernard COUSIN (*IRISA*)
Dr. Ali BOUDANI (*IRISA*)

Soutenu le 27 Juin 2006
Devant le jury d'examen

Mr. Farouk KAMOUN	Professeur à l'ENSI	Président
Mr. Slim MHIRI	Maître assistant à l'ENSI	Membre
Mr. Abdelfettah BELGHITH	Professeur à l'ENSI	Encadrant

Dédicaces

A mes parents
A mes frères
A Mekki
A tous mes amis
A tous ceux que j'aime
Je dédie ce travail

Remerciements

Toute ma gratitude à M. Abdelfettah BELGHITH, mon directeur de stage à l'ENSI, pour l'attention qu'il a accordé au bon déroulement de ce travail.

Je tiens également à remercier M. Bernard COUSIN, mon directeur de stage à l'IRISA, pour m'avoir accueilli au sein de l'équipe ARMOR et pour avoir suivi de très près l'évolution de ce mémoire.

Je remercie très vivement M. Ali BOUDANI, mon encadreur direct à l'IRISA, pour l'aide précieux qu'il m'a fourni tout au long de ce stage.

J'aimerais aussi remercier Mr. Farouk KAMOUN et Mr. Slim MHIRI pour avoir accepté de juger mon travail.

Mes plus tendres pensées sont à mes parents Sadok et Rafika, à mes frères Hichem, Ghassen et Naoufel et à mon fiancé Mekki, pour tout le soutien moral qu'ils m'ont procuré.

Mes remerciements sont également à toute personne ayant contribué de près ou de loin au bon déroulement de ce stage.

Table des matières

Table des figures

Introduction

Contexte

Les applications courantes d'Internet fonctionnent pour une grande majorité dans le mode *unicast* IP (point à point), selon lequel un poste émetteur envoie un paquet destiné à un poste récepteur. Le réseau transporte le paquet vers ce destinataire, et uniquement vers lui.

L'émergence de nombreux services multimédia fondamentalement basés sur des communications de groupe, notamment la diffusion temps-réel de vidéo, d'audio ou d'informations financières, les vidéoconférences et les espaces virtuels distribués, a favorisé l'évolution vers un nouveau mode de communication IP : le *multicast* IP, permettant de gérer efficacement des communications en mode multipoint. En *multicast*, la source émet des paquets avec une adresse de destination qui est en fait un identifiant de groupe. Au niveau de ses routeurs IP, le réseau démultiplie les paquets de telle sorte que tous les postes récepteurs abonnés à ce groupe en reçoivent chacun une copie. Le *multicast* permet de consommer moins de bande passante et diminue la charge des processeurs des stations émettrices puisque ces dernières ne doivent émettre qu'un seul flux de données. C'est une technologie vouée à un grand avenir à l'heure où l'on parle de radio et même de télévision sur Internet. Ces diffusions se font aujourd'hui en *unicast* ce qui cause une grande consommation de bande passante et qui nécessite des serveurs audio et vidéo de grande puissance.

Par ailleurs, l'arrivée de nouvelles technologies de transmission de données sur des réseaux sans fil comme 802.11, GPRS, UMTS, a permis le développement et l'amplification des nouveaux services de mobilité. La révolution dans le domaine de la mobilité des hôtes IP s'est récemment suivie par des recherches innovantes dédiées au sujet de la mobilité des réseaux dans la nouvelle génération d'Internet (NEMO : *Network Mobility*). Un réseau mobile est un réseau qui change dynamiquement son point d'ancrage à la topologie d'Internet. Les réseaux déployés dans les véhicules (VAN *: Vehicular Area Network*) et les réseaux personnels (PAN : *Personal Area Network*) sont des exemples typiques de réseaux mobiles. Dans ce cadre, le groupe NEMO de l'IETF a standardisé un support de base pour la mobilité des réseaux [RFC3963]. Ce support est une extension de Mobile IPv6 [RFC3775], le standard IETF pour la gestion des nœuds mobile dans IPv6, nouvelle génération du protocole IP [RFC2460]. Le support de base de NEMO permet aux nœuds d'un réseau mobile de rester joignables et de garder la continuité de leurs communications *unicast* en

cours alors que le réseau est entrain de se déplacer.

Or l'apport reconnu du *multicast* IP pour les communications de groupe en matière de conservation de la bande passante reste important dans les environnements mobiles. Plusieurs approches ont été alors proposées pour permettre l'association entre mobilité et *multicast*. Cependant, dédiées au support des nœuds *multicast* mobiles, ces approches s'avèrent insuffisantes pour fournir un service *multicast* aux réseaux mobiles. En effet, de nouvelles contraintes ont été introduites par la mobilité des réseaux, à savoir le besoin de garder cette mobilité transparente aux nœuds internes pouvant même être démunis de tout support de mobilité, et l'emboîtement de la mobilité selon lequel un nœud ou un réseau mobile vient s'attacher à un réseau étranger qui est lui-même mobile.

Dans le but de fournir un service *multicast* dans les environnements NEMO, les auteurs de [JAN04] proposent de combiner des approches classiques de gestion des nœuds *multicast* mobiles, à savoir l'enregistrement à distance et le tunneling bidirectionnel [BET00][XYL97], à la technique de *proxying* MLD, originellement proposée dans [FEN04] pour remplacer le déploiement d'un protocole de routage *multicast* dans un réseau de bordure à topologie simplifiée. Le *proxying* MLD a l'avantage d'être facile à mettre en place et d'offrir une transparence de la mobilité du réseau à ses nœuds *multicast*. Cependant, nous montrons qu'il présente des inconvénients majeurs allant de la redondance du trafic et la construction inutile de branches *multicast* au risque de création de boucles de transmission lorsqu'il est déployé dans un réseau NEMO.

Nous proposons alors une nouvelle approche basée sur l'utilisation de la technique de transmission *multi-unicast* explicite ou Xcast [BOI05], associée à l'enregistrement à distance et au tunneling bidirectionnel pour supporter respectivement les récepteurs et les sources *multicast* situés dans un réseau NEMO. Le *multi-unicast* explicite est un nouveau mode multipoint de transmission de données qui, à la différence du *multicast* IP traditionnel basé sur l'utilisation d'une adresse de groupe, utilise un encodage explicite de la liste des destinations dans les paquets à envoyer. Ceci lui permet de s'en passer de la construction d'un arbre de livraison lors d'une communication de groupe. Xcast offre une alternative intéressante quand il s'agit de gérer un très grand nombre de sessions multipoint de petite taille. Le protocole Xcast+ [MYU01] est une extension de la spécification de base de Xcast [BOI05] qui permet de supporter les hôtes *multicast* standard en utilisant des protocoles standard de gestion des groupes *multicast* (IGMPv3 [RFC3376] pour IPv4 et MLDv2 [RFC3810] pour IPv6). Notre solution utilise le protocole Xcast+6 [MYU01], la version de Xcast+ dédiée à IPv6, à l'intérieur du réseau mobile. Pour ce faire, nous spécifions la manière dont Xcast+6 doit être déployé à l'intérieur d'un réseau mobile. De plus, nous définissons un mécanisme d'optimisation permettant de gérer efficacement les communications multipoint intra-NEMO dans les réseaux mobiles emboîtés.

Plan du mémoire

Ce mémoire est organisé en cinq chapitres.

Dans le premier chapitre nous abordons l'état de l'art du *multicast* IP, à savoir ses notions de base, les algorithmes et les protocoles de routage *multicast*, et l'évolution du modèle du *multicast* IP.

Nous consacrons le deuxième chapitre à la description des protocoles de routage *multi-unicast* explicite. Nous y présentons le protocole Xcast, et ses extensions Xcast+ et GXcast (Generalized Xcast).

Le troisième chapitre porte sur la mobilité des réseaux dans IPv6. Nous commençons par une vue d'ensemble sur la gestion de la mobilité des nœuds par le protocole Mobile IPv6. Nous exposons ensuite la problématique posée par la mobilité des réseaux, et nous terminons par une description du fonctionnement général du support de base de NEMO.

Dans le quatrième chapitre, nous décrivons les enjeux de l'association entre *multicast* et mobilité dans IPv6, comprenant mobilité des nœuds et celle des réseaux. Nous nous intéressons en particulier à la solution basée sur le déploiement du *proxying* MLD dans les réseaux mobiles, ce qui nous permet de déceler les problèmes qu'elle présente.

Dans le sixième chapitre, nous proposons une nouvelle approche permettant de fournir un service *multicast* aux réseaux mobiles. Nous commençons par définir une solution de base qui ne tient pas en compte la question d'emboîtement de la mobilité. Nous optimisons ensuite cette solution pour qu'elle puisse supporter de façon efficace les réseaux mobiles emboîtés. Enfin, nous menons une évaluation théorique de notre proposition, en se basant sur différents critères tels que la création de boucles de transmission, l'optimalité des chemins, la transparence vis-à-vis des MNNs, etc.

Nous concluons ce mémoire par une synthèse du travail réalisé et une présentation des perspectives qu'il offre.

Chapitre 1

Le *Multicast* IP

La grande majorité d'applications courantes d'Internet, telles que le transfert de fichiers, le jeu en ligne et l'email, utilisent un routage *unicast* (point à point) selon lequel un paquet est conduit par le réseau d'un hôte émetteur (source) vers un hôte récepteur (destination). Un paquet IP *unicast* contient l'adresse du hôte source et l'adresse du hôte destination.

La transmission des données a évolué pour faire face à de nouveaux besoins. Elle est passée de la communication en mode *unicast* à la communication en mode *multicast* selon lequel l'envoi des données se fait de un vers plusieurs ou de plusieurs vers plusieurs. Ce type de transmission a été motivé par l'apparition des applications coopératives telles que la visioconférence, l'audioconférence, les applications de tableau blanc ou les jeux en réseau, et des applications de diffusion, telles que la TV ou la radio sur Internet. Basées sur une communication de groupe, de telles applications sont inefficacement gérées par l'*unicast* IP puisque des paquets multiples avec les mêmes données doivent être conduits aux différents destinataires. L'idée du *multicast* IP est alors de n'envoyer qu'un seul paquet par lien, celui-ci étant dupliqué par le réseau uniquement quand cela est nécessaire. Le *multicast* IP permet ainsi de minimiser la consommation de la bande passante en évitant le passage multiple d'un paquet sur un même lien (*cf.* figure 1.1).

Dans ce chapitre, nous commençons par présenter les fondements du *multicast* IP, à savoir la notion de groupe *multicast* et les composantes fondamentales du routage *multicast*. Nous détaillons ensuite les différentes techniques de routage *multicast*, puis nous décrivons les principaux protocoles qui ont été proposés. Nous terminons avec une description de l'évolution du modèle de service du *multicast* IP.

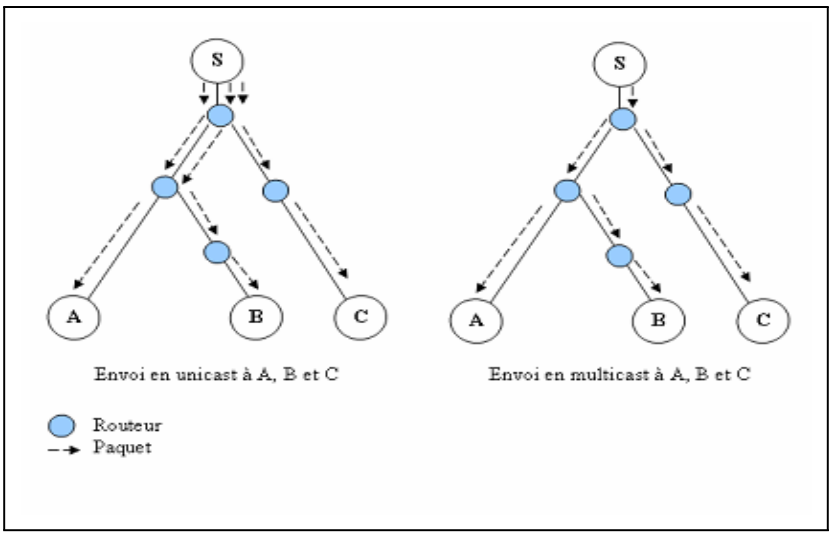


FIG. 1.1 – *Multicast* vs *unicast*

1.1 Les fondements du *multicast* IP

1.1.1 Notion de groupe *multicast*

Le *multicast* IP a été présenté pour la première fois par Steve Deering en 1988 sous le nom de modèle Hôte Groupe (*Host Group Model*) [DEE91]. Ce modèle permet la livraison multiple des données de plusieurs à plusieurs : une ou plusieurs sources envoient vers un ensemble de destinataires. Chaque paquet est envoyé une seule fois, puis dupliqué dans le réseau chaque fois que nécessaire. Ceci permet de réduire la charge des émetteurs ainsi que la charge globale du réseau.

Le modèle se base sur la notion abstraite de groupe *multicast*, qui est une destination logique identifiée par un nouveau type d'adresse dite adresse de groupe ou adresse *multicast*. Lorsqu'une source envoie un paquet destiné à un groupe *multicast*, ce paquet est conduit par le réseau jusqu'aux différents récepteurs abonnés à ce groupe appelés membres du groupe. Notons qu'une source *multicast* n'a pas à se soucier de connaître les adresses des différents membres, seule l'adresse du groupe lui est nécessaire.

La notion de groupe *multicast* dans le modèle de Deering est fondée sur les règles suivantes :

- Un paquet envoyé à un groupe est reçu par tous les membres de ce groupe.
- On doit être un membre d'un groupe *multicast* pour recevoir les données qui lui sont envoyés.
- Le groupe est ouvert : il est possible de transmettre des données vers le groupe sans en être membre.
- Le groupe est dynamique : un client peut rejoindre ou quitter le groupe à n'importe quel moment. L'existence du groupe est donc indépendante de celle des

différents membres.
- Il n'y a pas de restriction topologique sur les membres : un membre peut être placé n'importe où dans le réseau.

Il est à noter que le modèle autorise l'appartenance simultanée d'un hôte à plusieurs groupes *multicast*.

Dans IPv4, un groupe est identifié et accessible par une adresse de classe D (dans la plage de 224.0.0.0 à 239.255.255.255). En IPv6, Les adresses *multicast* se voient assigner le préfixe ff00 : :/8. Une adresse *multicast* ne peut être que destinataire : les sources ont toujours une adresse *unicast*.

1.1.2 Composantes du routage *multicast*

Dans le *multicast* IP, il y a séparation entre deux composantes : la gestion des groupes et le routage des paquets *multicast* (*cf.* figure 1.2).

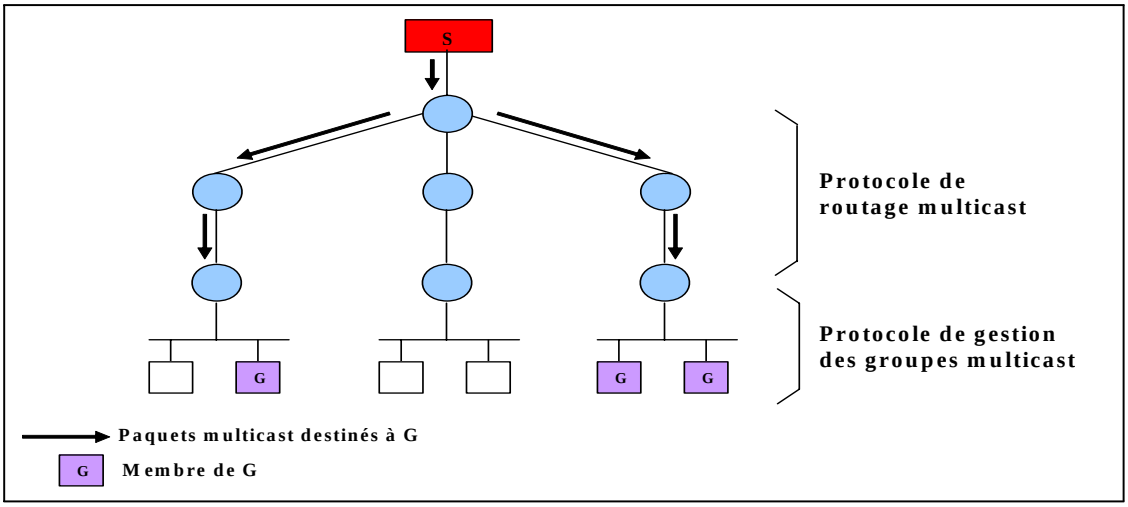


FIG. 1.2 – Composantes du routage *multicast*

La gestion des groupes *multicast* est assurée par un protocole déroulé au niveau du lien local. Un tel protocole permet à un routeur de découvrir, pour chaque groupe *multicast*, la présence de récepteurs sur ses liens directement attachés. Il s'agit alors d'un protocole de type hôte-routeur. Les protocoles IGMP (*Internet Group Management Protocol*) [RFC2236][RFC3376], et MLD (*Multicast Listener Discovery*) [RFC2710][RFC3810] assurent la gestion des groupes *multicast* respectivement dans IPv4 et IPv6. Nous présentons le protocole MLD plus loin dans ce chapitre (*cf.* section 1.3.1).

Au niveau du Backbone, un protocole de routage *multicast* est nécessaire à l'acheminement des paquets *multicast*. Un tel protocole, de type routeur-routeur, assure la duplication de chaque paquet *multicast* autant de fois que nécessaire pour le faire arriver aux différents récepteurs concernés.

1.2 Les techniques de routage *multicast*

Plusieurs algorithmes sont potentiellement utilisables par les protocoles de routage *multicast*, permettant l'acheminement des paquets *multicast* de la source vers les récepteurs concernés lorsque ceux-ci ne sont pas situés sur le même LAN que la source.

Ces algorithmes peuvent être classées en trois ensembles, selon la technique utilisée :

- Les techniques primitives
- La technique d'arbre basé à la source
- La technique d'arbre partagé

L'inondation et l'arbre *multicast* recouvrant sont deux algorithmes de routage *multicast* primitifs. Ils ont été qualifiés ainsi à cause de leur tendance à gaspiller la bande passante ou les ressources des routeurs dans le réseau. De plus, ils présentent un problème de passage à l'échelle avec l'augmentation de la taille du réseau ou du nombre des sources et groupes *multicast*.

Vu l'inefficacité des techniques de routage *multicast* primitives et leur non résistance au facteur d'échelle, des techniques plus élaborées ont vu le jour, ayant pour but de construire des arbres de livraison permettant une meilleure utilisation des ressources réseau. Une première approche consiste à construire un arbre basée (enraciné) à la source (SBT : *Source Based Tree*). Une deuxième approche est celle de construire un arbre partagé entre toutes les sources envoyant vers un groupe *multicast* donné (ST : *Shared Tree*).

1.2.1 Techniques primitives

1.2.1.1 L'algorithme d'inondation

Le moyen le plus simple d'envoyer des paquets en mode multipoint est d'implémenter un algorithme d'inondation (*flooding*) dont le principe est le suivant : un routeur qui reçoit un paquet destiné à un groupe *multicast* vérifie d'abord si c'est la première fois qu'il reçoit ce paquet ou s'il s'agit d'une répétition. Si c'est la première réception, il duplique le paquet sur toutes ses interfaces autres que l'interface de réception. Sinon, il détruit le paquet. Cette procédure garantit à chaque paquet *multicast* de visiter tout les routeurs du réseau sans génération de boucle de transmission (*cf.* figure 1.3).

L'algorithme d'inondation est très simple à être déployé puisque les routeurs ont seulement besoin de garder la trace des paquets *multicast* récemment reçus, et n'ont pas à maintenir une table de routage *multicast*. Cependant, l'inondation est particulièrement consommatrice de bande passante et ne passe pas à l'échelle avec l'augmentation de la taille du réseau car elle génère un grand nombre de paquets dupliqués et utilise tous les chemins possibles. De plus, les routeurs doivent conserver une grande quantité d'informations mémorisant l'historique de réception des paquets *multicast* afin d'éviter la rediffusion des paquets déjà reçus.

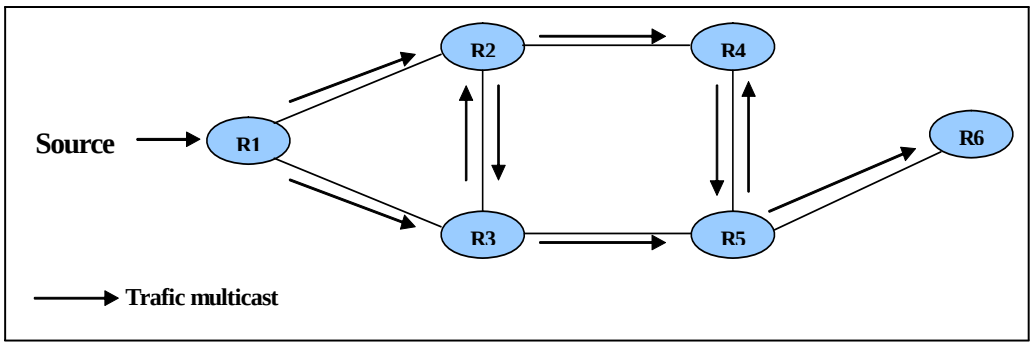


FIG. 1.3 – Algorithme d'inondation

1.2.1.2 L'algorithme d'arbre *multicast* recouvrant

Une solution plus efficace que l'inondation est celle de construire un arbre de livraison qui recouvre le réseau afin de garantir que toutes les destinations possibles soient atteintes par chaque flux *multicast*. Un arbre recouvrant (*Spanning Tree*) est un arbre sur lequel un chemin unique relie chaque couple de routeurs dans le réseau. La construction de l'arbre est préalable, elle se fait indépendamment des sources et des récepteurs *multicast*. Un tel arbre, commun à tous les flux *multicast* circulant dans le réseau, est enraciné en un routeur particulier du réseau appelé pont principal. L'arbre relie le pont principal à chacun des routeurs du réseau selon un plus court chemin. La figure 1.4 illustre la transmission de trafic *multicast* provenant de deux sources différentes sur un arbre recouvrant. Le routeur R2 a été choisi comme pont principal pour cet arbre.

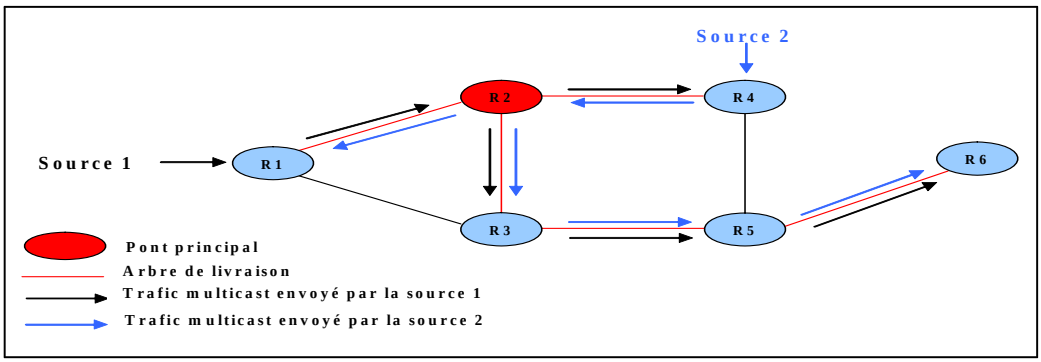


FIG. 1.4 – Algorithme de l'arbre *multicast* recouvrant

Chaque routeur dans le réseau maintient, pour chacune de ses interfaces, l'information "passante" si cette interface fait partie de l'arbre recouvrant ou "bloquante" sinon. Lorsqu'il reçoit un paquet *multicast*, il le duplique sur toutes ses interfaces passantes à l'exception de l'interface sur laquelle le paquet est arrivé. La structure d'arbre recouvrant permet de desservir tous les routeurs du réseau en trafic *multicast*

sans risque de création de boucles de transmission.

En limitant la transmission des paquets *multicast* aux branches de l'arbre, cet algorithme permet d'éviter la saturation de tous les chemins dans le réseau. Cependant, il surcharge un sous ensemble de liens en trafic *multicast*, et ne permet pas de produire des chemins optimaux entre chaque source *multicast* et ses destinataires. Ces inconvénients, associés à la difficulté de construction d'un arbre recouvrant dans les topologies larges et compliquées, rendent l'algorithme non applicable à grande échelle.

1.2.2 Technique d'Arbre basé à la source

Selon la technique d'arbre basé à la source, le chemin entre une source *multicast* et chaque destinataire est calculé de façon à optimiser un coût prédéfini (par exemple le délai de transmission), permettant ainsi de construire un arbre de plus court chemin (*Shortest Path Tree*). Pour ce faire, une technique de transmission appelée "relais sur le chemin inverse" ou RPF (*Reverse Path Forwarding*) est utilisée. Différents algorithmes basés sur le RPF ont été définis, à savoir l'algorithme RPB (*Reverse Path Broadcasting*), l'algorithme TRPB (*Truncated Reverse Path Broadcasting*) et l'algorithme RPM (*Reverse Path Multicasting*). TRPB et RPM apportent des améliorations à RPB, algorithme de base de la famille RPF.

1.2.2.1 L'algorithme RPB

L'algorithme RPB (*Reverse Path Broadcasting*) est l'algorithme de base de construction d'un arbre basé à la source. Il est relativement simple.

Lors de la réception d'un paquet *multicast*, un routeur compare l'interface sur laquelle ce paquet est arrivé à l'interface menant vers la source par le plus court chemin (appelée interface entrante ou interface parent). S'il trouve que c'est la même interface, il duplique le paquet sur toutes ses interfaces autres que celle de réception (interfaces sortantes ou interfaces filles). Sinon, il détruit le paquet. Ce test, appelé test RPF (*RPF check*), est réalisé par une simple consultation de la table de routage *unicast*. La figure 1.5 illustre la transmission d'un paquet *multicast* selon l'algorithme RPB.

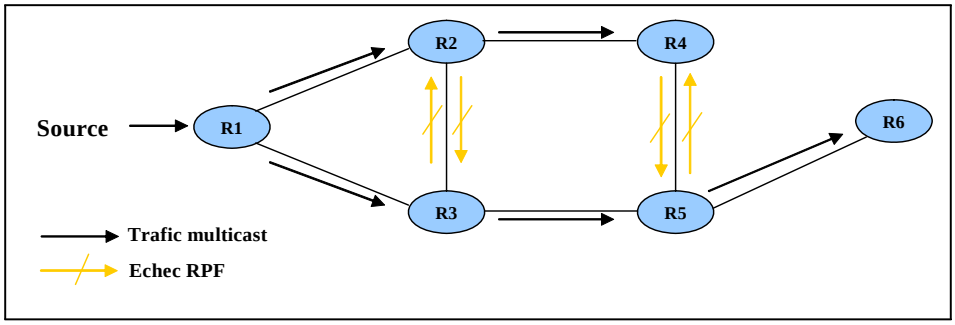


FIG. 1.5 – L'algorithme RPB

Il est à noter que l'algorithme RPB de base, tel que spécifié ci-dessus, n'est qu'une amélioration de l'algorithme d'inondation permettant de s'en passer de la mémorisation de l'historique des réceptions des paquets *multicast*. En effet, la redondance du trafic sur les liens n'est pas évitée (*cf.* figure 1.5).

Toutefois, l'algorithme RPB peut être amélioré en réduisant le nombre de paquets inutilement dupliqués dans le réseau. Pour ce faire, un routeur ne relaie un paquet *multicast* que vers ses routeurs fils se trouvant en aval sur l'arbre de livraison, c'est-à-dire ceux pour lesquels le plus court chemin vers la source passe par le routeur en question. En effet, les autres routeurs fils rejetteraient un tel paquet. La figure 1.6 illustre la transmission d'un paquet *multicast* selon l'algorithme RPB amélioré sur le même réseau pris comme exemple.

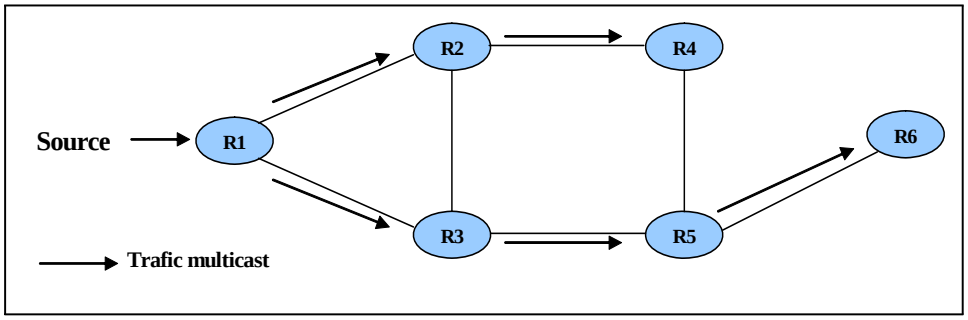


FIG. 1.6 – L'algorithme RBP amélioré

En remarquant qu'il n'est pas sur le plus court chemin de R3 vers la source, R2 ne relaie pas le paquet à R3. De même, R3, R4 et R5 ne relaient pas le paquet respectivement à R2, R5 et R4.

Cette amélioration de RPB requiert de chaque routeur d'être en mesure de savoir s'il se trouve sur le chemin le plus court que chacun de ses voisins aurait choisi pour joindre la source du paquet qu'il s'apprête à leur relayer. Une telle information peut être facilement déduite si un protocole de routage *unicast* à état de liens est utilisé, puisque chaque routeur aurait une vision globale de la topologie de tout le domaine. Si, par contre, un protocole à vecteur de distance est employé, cette information doit être échangée entre routeurs voisins.

L'algorithme RPB est simple à être déployé, il ne requiert ni une gestion globale de l'arbre de livraison, ni une mémorisation de l'historique des réceptions. De plus, il construit un arbre recouvrant de plus courts chemins si les liens du réseau sont symétriques (un lien est dit symétrique s'il admet le même coût dans les deux sens). Par ailleurs, la construction d'un arbre différent par source *multicast* permet d'équilibrer la charge des liens du réseau en trafic *multicast*, et offre donc une meilleure utilisation des ressources réseau comparée à l'algorithme d'arbre recouvrant unique (*cf.* section 1.2.1.2).

L'inconvénient majeur de RPB provient du fait qu'il ne prend pas en compte les informations d'appartenance aux groupes *multicast* lors de la construction d'un arbre

de livraison. Il en résulte une transmission inutile de paquets *multicast* vers des LANs n'hébergeant aucun membre concerné.

1.2.2.2 L'algorithme TRPB

L'algorithme TRPB (*Truncated Reverse Path Broadcasting*) a été conçu pour venir à bout des limitations de l'algorithme RPB. Comme ce dernier, l'algorithme TRPB produit un arbre de livraison basé à la source. La seule différence entre ces deux algorithmes provient du fait que dans TRPB, les routeurs utilisent les états installés par le protocole de gestion des groupes *multicast*. Ces états permettent aux routeurs de découvrir la présence d'ordinateurs hôtes abonnés à des groupes sur chacun des LAN qu'ils connectent au reste de l'arbre *multicast* et de connaître l'identité de ces groupes (*cf.* section 1.3.1). Un routeur évite ainsi de relayer des paquets dans un LAN feuille (c'est-à-dire auquel aucun autre routeur n'est attaché) où il n'existe aucun abonné aux groupes *multicast* en cours : la branche associée est tronquée. On dit alors que les routeurs élaguent l'arbre de livraison de ses feuilles qui n'ont pas lieu d'être. Ainsi, un arbre de livraison est construit par couple (S,G). Sur l'exemple de la figure 1.7, le routeur R4 effectue un élagage du LAN feuille qu'il dessert, puisque ce LAN n'héberge aucun membre actif de G.

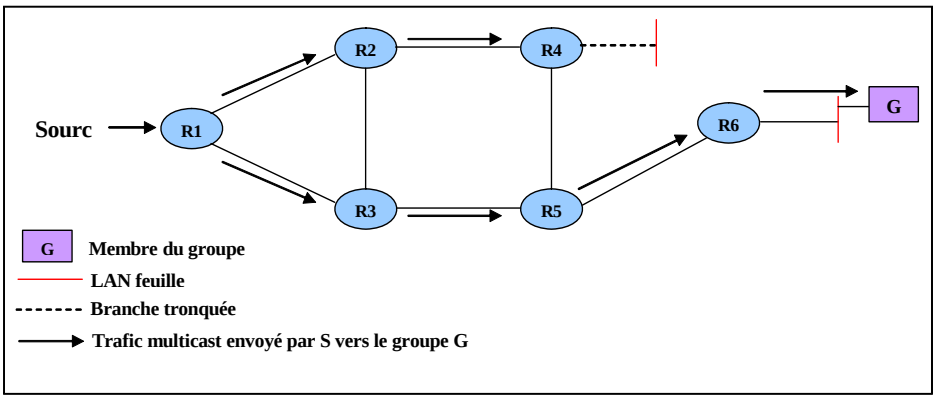


FIG. 1.7 – L'algorithme TRPB

Il est à noter que deux arbres TRBP basés en une même source et relatifs à deux groupes *multicast* différents coïncident sauf en leurs feuilles.

L'algorithme TRPB améliore l'algorithme RBP mais ne résout qu'en partie les problèmes que présente ce dernier. En effet, l'élagage effectué au niveau des feuilles ne permet pas d'éliminer la transmission des paquets *multicast* sur les branches inutiles, c'est à dire ne menant à aucun membre *multicast* (*cf.* figure 1.7).

1.2.2.3 L'algorithme RPM

L'algorithme RPM (*Reverse Path Multicasting*) est une amélioration apportée aux algorithmes RPB et TRPB. Comme TRPB, RPM crée un arbre de livraison *multicast*

différent par couple (source, groupe *multicast*) en se basant sur les informations d'appartenance aux groupes *multicast*. Alors que TRPB se contente d'élaguer les LAN feuilles n'hébergeant aucun membre intéressé, RPM élimine les branches inutiles dans leur totalité pour ne conserver que les branches reliant la source aux membres actifs, et ce grâce à des messages d'élagage dits messages *prune* échangés entre les routeurs.

RPM opère de la manière suivante : lorsqu'un routeur reçoit de S un premier paquet *multicast* dont la destination est G, il le relaye le long d'un arbre de livraison identique à celui que construit l'algorithme TRPB pour (S,G) (*cf.* section 1.2.2.2). Ce premier paquet est donc reçu par l'ensemble des routeurs feuilles de l'arbre TRPB (c'est-à-dire les routeurs les plus en aval). En se basant sur les états localement installés par le protocole de gestion des groupes, les routeurs feuilles découvrent la présence d'ordinateurs hôtes abonnés à G sur chacun des réseaux physiques qu'ils connectent au reste de l'arbre TRPB. Un routeur feuille qui ne connecte aucun membre de G (sur ses différents LAN fils) retourne un message *prune* sur son interface entrante dans l'arbre TRPB. Il incite ainsi le routeur immédiatement situé en amont à ne plus relayer de paquets relatifs à (S,G) sur l'interface sortante par laquelle ce dernier reçoit le message *prune*. Si le routeur amont n'a aucun membre de G connecté à ses LANs feuilles, et qu'il reçoit un message *prune* sur toutes ses autres interfaces sortantes, il génère à son tour un message *prune* relatif à (S,G). Ces opérations sont illustrées la figure 1.8 (a).

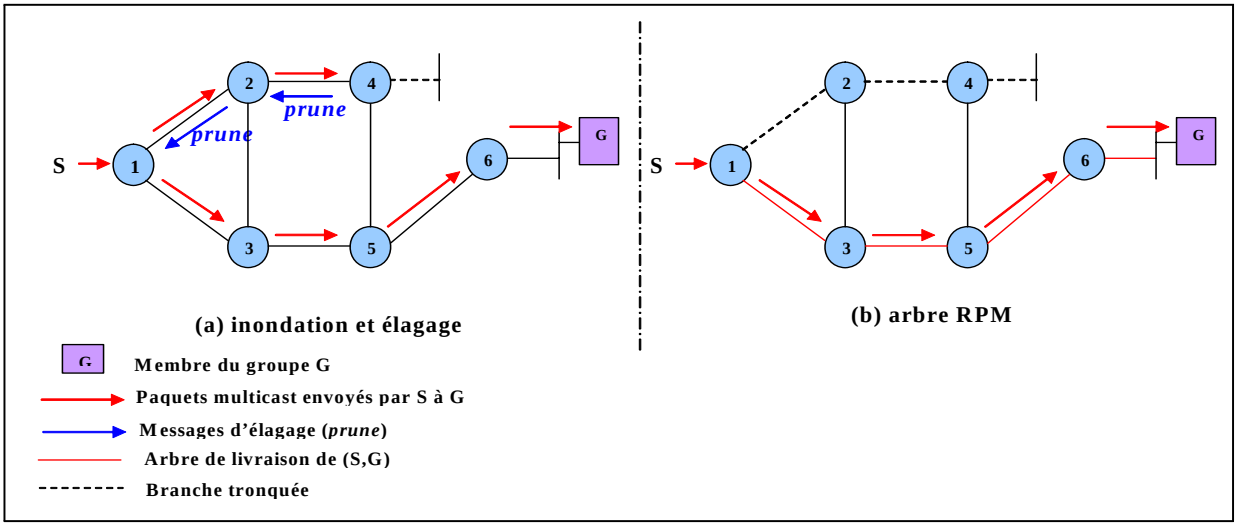


FIG. 1.8 – L'algorithme RPM

Le processus de génération des messages *prune* permet d'élaguer toutes les branches inutiles de l'arbre TRPB : les branches qui subsistent à ce processus sont celles qui permettent à la source de joindre les seuls LANs qui hébergent des membres actifs de G (*cf.* figure 1.8 (b)).

Comme l'appartenance aux groupes et la topologie du réseau peuvent changer dynamiquement, l'état d'élagage de l'arbre doit être régulièrement mis à jour. Pour

ce faire, l'algorithme RPM répète la procédure d'inondation/élagage de façon périodique : à des intervalles réguliers, l'information d'élagage expire dans la mémoire de tous les routeurs, et le prochain paquet est alors conduit vers tous les routeurs en aval (*cf.* figure 1.8 (a)).

Notons qu'à la différence des algorithmes précédents de la famille RPF (RPB et TRPB) dont l'arbre de livraison est implicite, l'algorithme RPM maintient une table de routage *multicast* explicite.

Malgré les améliorations apportées par l'algorithme RPM, ce dernier présente encore plusieurs problèmes. En effet, la procédure périodique d'inondation/élagage génère un trafic de gestion important, et est particulièrement consommatrice de bande passante. Par ailleurs, l'algorithme ne permet pas une utilisation efficace des ressources mémoire puisque chaque routeur du réseau doit maintenir, par couple (S,G), soit une entrée relative à (S,G) dans sa table de routage, soit un état d'élagage pour (S,G). Ceci surcharge les routeurs et risque de générer des tables de routage *multicast* de taille très importante. Ces limitations réduisent considérablement les performances de cet algorithme quand il s'agit d'un réseau de grande taille ou contenant un nombre important de sources et de groupes *multicast*.

1.2.3 Technique d'arbre partagé

Dans le but de fournir un routage *multicast* qui puisse passer à l'échelle avec l'augmentation de la taille du réseau et du nombre des sources et des groupes *multicast* actifs, des algorithmes plus récents ont été proposés basés sur une nouvelle approche : la technique d'arbre partagé (ST : *Shared Tree*). Contrairement aux algorithmes SBT qui construisent un arbre par source ou par couple (source, groupe), les algorithmes ST construisent un arbre de livraison unique par groupe *multicast*, partagé entre toutes les sources envoyant vers ce groupe : le trafic *multicast* destiné à un groupe est transmis sur ce même arbre partagé quelque soit sa provenance. L'approche d'arbre partagé diffère de l'algorithme d'arbre recouvrant (*cf.* section 1.2.1.2) par le fait qu'elle établit un arbre de livraison différent pour chaque groupe *multicast*. Les stations qui veulent recevoir le trafic destiné à un groupe donné doivent rejoindre explicitement l'arbre de livraison associé à ce groupe.

La transmission des paquets *multicast* s'effectue via un routeur spécifique appelé *core*, sélectionné dès la configuration du réseau. L'adhésion d'un nouveau membre à G provoque la construction d'une branche d'arbre relative à G allant du routeur *core* jusqu'à ce nouveau membre. Les états de routage maintenus par les routeurs situés sur cette branche sont de la forme (*,G), où * désigne une source quelconque envoyant vers le groupe G. Lorsqu'une source envoie un paquet *multicast* destiné à G, son routeur *multicast* local encapsule ce paquet dans un paquet *unicast* qu'il envoie vers le *core*, qui le désencapsule et l'émet en mode *multicast* sur l'arbre de livraison. Le paquet est alors transmis de routeur en routeur sur l'arbre jusqu'aux stations abonnées à G. Sur l'exemple représenté par la figure 1.9, les deux sources 1 et 2 utilisent le même arbre partagé pour envoyer vers le groupe *multicast* G. Le nœud R2 joue le rôle de *core* pour cet arbre.

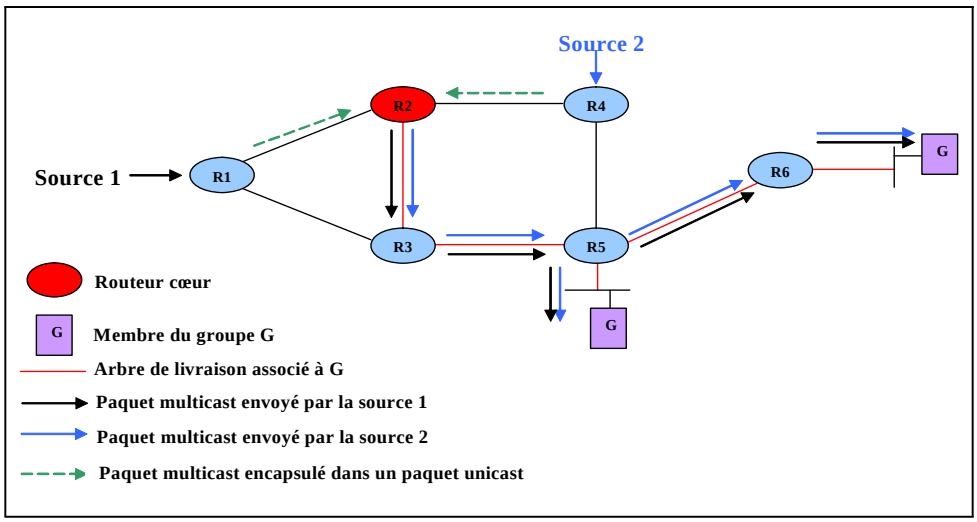


FIG. 1.9 – Technique de l'arbre partagé

La technique d'arbre partagé permet un meilleur passage à l'échelle que celle d'arbre basé à la source grâce à une gestion plus efficace des ressources du réseau. D'une part, seuls les routeurs situés sur un arbre doivent maintenir une information de routage relative à cet arbre. D'autre part, un seul état de routage est maintenu par groupe *multicast* quelque soit le nombre de sources envoyant vers ce groupe, ce qui réduit considérablement la taille des tables de routage *multicast*. Par ailleurs, en remplaçant la procédure d'inondation périodique par celle d'abonnement explicite des membres, la technique d'arbre partagé permet de conserver la bande passante des liens puisqu'elle réduit le trafic de gestion : seul le trafic utile circule sur les liens.

Cependant, cette technique présente certains inconvénients. En effet, l'utilisation d'un routeur *core* entraîne une concentration de trafic, voire une congestion au voisinage de ce routeur. De plus, le *core* constitue un point de vulnérabilité critique dont dépend tout le groupe. Par ailleurs, les chemins sur un arbre partagé ne sont pas optimaux, ce qui augmente les délais de transmission des paquets *multicast*. Mais il est à noter que même pour un arbre basé à la source, les chemins peuvent ne pas être optimaux, puisqu'il s'agit des meilleurs chemins inverses, qui ne sont pas forcément les meilleurs chemins menant de la source vers les destinataires (cas d'un réseau dont les chemins sont asymétriques).

1.3 Les protocoles *multicast*

Trois types de protocoles sont utilisés pour le déploiement du *multicast* sur Internet.

Au niveau lien local, le protocole de gestion des groupes *multicast* réalise la signalisation entre l'hôte et son routeur d'accès à l'Internet. Il est utilisé par un routeur de bordure pour découvrir la présence de récepteurs *multicast* sur ses liens directement

attachés, ainsi que les adresses *multicast* concernées. Les protocoles IGMP (*Internet Group Management Protocol*) [RFC2236][RFC3376], et MLD (*Multicast Listener Discovery*) [RFC2710][RFC3810] sont les standards IETF permettant la gestion des groupes *multicast* respectivement dans IPv4 et IPv6.

Le second type de protocole, dont la portée est limitée à un même domaine d'administration, est appelé protocole intra-domaine ou MIGP (*Multicast Interior Gateway Protocol*). Il permet la construction d'un arbre de livraison limité à un domaine. Les protocoles DVMRP [RFC1075], MOSPF [RFC1247][RFC1584], PIM [ADA03][RFC2362] et CBT [RFC2189][RFC2201] représentent des exemples de ce type de protocole.

Enfin, des protocoles inter-domaines sont employés par les routeurs de bordure pour permettre le service *multicast* à travers différents systèmes autonomes. Les protocoles BGMP [THA03], MBGP [RFC2283] et MDSP [RFC3618] en sont des exemples.

1.3.1 Le protocole de gestion des groupes *multicast* MLD

La gestion des groupes *multicast* est assurée en IPv6 par le protocole MLD (*Multicast Listener Discovery*). Ce protocole existe en deux versions : MLDv1 [RFC2710] et MLDv2 [RFC3810]. Ces versions sont les équivalents respectifs de IGMPv2 [RFC2236] et IGMPv3 [RFC3376] pour IPv4. Comparé à MLDv1, MLDv2 offre le filtrage de sources *multicast*, qui permet à un hôte de choisir, pour chaque groupe *multicast* dont il est membre, les sources dont il veut recevoir le trafic.

1.3.1.1 MLD version 1

Version originale du protocole MLD, MLDv1 est aussi appelée MLD tout court. Tout comme IGMP, MLD est un protocole asymétrique qui spécifie un comportement différent pour les hôtes et les routeurs *multicast*. Il permet aux hôtes de déclarer leur appartenance à un ou plusieurs groupes auprès du routeur *multicast* sur leur lien IPv6 local, spontanément ou après interrogation du routeur. Les messages utilisés par le protocole MLD sont détaillés dans l'annexe A.

Le routeur MLD interroge régulièrement les hôtes sur son lien local sur leur appartenance aux groupes *multicast* en envoyant des messages de recensement général à l'adresse de diffusion générale sur le lien (FF02 : :1). Les hôtes utilisent alors des messages dits rapports d'abonnement MLD pour répondre aux recensements du routeur. Ces interrogations et réponses périodiques permettent au routeur de maintenir une table à jour d'appartenance aux groupes. Pour éviter que toutes les réponses arrivent en même temps au routeur, chaque hôte patiente durant une période aléatoire avant de répondre. Si un hôte répond pour un groupe, tous les autres hôtes annulent leurs réponses concernant ce même groupe. Le routeur ne conserve donc pas la liste de tous les membres, mais plutôt celle des groupes pour lesquels il existe au moins un abonné dans son réseau local. Ceci est bien justifié car il suffit qu'il y ait au moins un destinataire appartenant à un groupe sur le lien local pour que le routeur continue à

envoyer le trafic *multicast* correspondant. Pour cesser d'appartenir à un groupe *multicast* donné, un hôte peut simplement ne plus répondre aux messages de recensement du routeur par des rapports d'abonnement concernant ce groupe. S'il s'avère que le hôte était le dernier membre du groupe en question sur le lien, l'état du routeur pour ce groupe expire après un laps de temps, et les paquets *multicast* correspondants ne sont alors plus diffusés sur le lien.

Dans le cas où plusieurs routeurs *multicast* sont sur le même lien local, un mécanisme d'élection est utilisé pour choisir le routeur recenseur. Celui-ci sera le seul responsable pour l'envoi des messages de recensement.

A côté des recensements périodiques, le protocole MLD permet aux hôtes de notifier leurs changements d'état sans attendre la prochaine interrogation du routeur recenseur, et ce grâce à deux types de messages MLD non-sollicités (*cf.* annexe A) :

- Pour s'abonner à un groupe *multicast* spécifique, un hôte envoie un rapport d'abonnement non-sollicité concernant ce groupe.
- La résiliation rapide est aussi une possibilité offerte par MLDv1, grâce à un message de résiliation d'abonnement. Le routeur recenseur répond avec un message de recensement spécifique à l'adresse du groupe en question. S'il n'y a plus de récepteur pour répondre à ce recensement, le routeur efface l'adresse *multicast* de sa table de routage.

1.3.1.2 MLD version 2

MLDv2 [RFC3810] est la nouvelle version du protocole de gestion des groupes *multicast* dans IPv6. Elle implante les fonctionnalités du protocole IGMPv3 défini pour IPv4, la plus importante étant l'introduction du filtrage des sources. Un hôte peut désormais spécifier les sources qu'il veut ou qu'il ne veut pas écouter pour une adresse *multicast* donnée. Cette information peut être utilisée par les protocoles de routage *multicast* afin d'éviter l'acheminement des paquets *multicast* provenant de certaines sources vers des liens où il n'y a pas de récepteur intéressé.

Il existe deux types de messages MLDv2 : les messages de recensement des récepteurs *multicast* et les rapports d'abonnement *multicast* version 2. Un message de recensement est soit général, soit spécifique à une adresse *multicast*, soit spécifique à la fois à une adresse et à une source *multicast* (*cf.* annexe A). Pour garder l'interopérabilité avec la version précédente de MLD, les messages de rapport d'abonnement *multicast* version 1 et de résiliation d'abonnement *multicast* sont également supportés.

Le filtrage des sources peut se faire à travers les rapports d'abonnements MLDv2 de deux façons différentes : en mode inclusion ou en mode exclusion. En mode inclusion, un hôte demande à ne recevoir le trafic *multicast* pour un groupe donné qu'en provenance d'une ou de plusieurs sources qu'il spécifie explicitement. En mode exclusion, un hôte demande, au contraire, de recevoir le trafic *multicast* pour un groupe donné sauf s'il provient de l'une des sources de la liste qu'il exclue.

Comme dans le cas du MLDv1, s'il existe plusieurs routeurs *multicast* sur le même lien local, un seul routeur recenseur est élu. Le recenseur envoie régulièrement des

messages de recensement général auxquels les récepteurs répondent avec des rapports d'abonnement contenant des enregistrements d'état actuel. Chaque hôte, ainsi que chaque routeur, gardent un état contenant le mode de filtrage et une liste des sources pour chaque adresse *multicast*. Dans le cas d'un hôte, ce sont les adresses *multicast* sur lesquelles il écoute. Dans le cas d'un routeur, ce sont les adresses *multicast* que ses récepteurs écoutent.

Un hôte demande la modification du mode de filtrage ou de la liste des sources pour un groupe *multicast* donné par le biais d'un rapport d'abonnement non-sollicité. A la réception de ce rapport, le routeur met à jour son état pour le groupe *multicast* concernée comme suit : si, suite à ces changements, il constate qu'une certaine source ne doit plus être acceptée, le routeur envoie un message de recensement spécifique pour cette source, afin de vérifier l'existence d'éventuels membres du groupe qui souhaitent toujours l'écouter. Un temporisateur est déclenché, et s'il expire sans que le routeur ait reçu un rapport d'abonnement concernant la source, celle-ci est éliminée de l'état local du routeur (pour le groupe en question). Si le routeur détecte qu'il ne reste plus de source sollicitée pour un certain groupe, il envoie un message de recensement spécifique pour ce groupe. Si des rapports d'abonnement concernant le groupe en question ne sont pas reçus en temps dû, l'adresse de ce groupe est effacée de l'état du routeur.

Notons enfin qu'un rapport d'abonnement MLDv2 avec une liste de sources vide équivaut à un rapport d'abonnement MLDv1 s'il est en mode exclusion (demander d'écouter le trafic destiné au groupe en provenance de toutes les sources), et à une résiliation MLDv1 s'il est en mode inclusion (n'écouter aucune source envoyant vers le groupe).

1.3.2 Les protocoles de routage *multicast* intra-domaine

Les LANs qui supportent le *multicast* peuvent se grouper en domaines afin d'échanger entre eux du trafic *multicast*. Un domaine dans l'Internet, aussi appelé système autonome est un ensemble d'équipements (routeurs, hôtes, liaisons...) sous la gestion d'une seule administration. Un domaine propose des services et est souvent sous la direction d'un opérateur ou d'une université. Il doit pouvoir rester le plus indépendant possible pour son administration et est souvent d'une taille très réduite par rapport à celle de l'Internet.

Le routage *multicast* intra-domaine nécessite la définition d'un arbre qui indique les chemins que les paquets doivent suivre afin d'atteindre leurs destinataires. L'élaboration de l'arbre de livraison nécessite un échange, entre les routeurs, de données de gestion s'appuyant sur des protocoles précis. Les arbres de livraison se modifient en permanence en fonction des stations émettrices et réceptrices qui quittent ou s'abonnent à tel ou tel groupe. On peut actuellement répartir les protocoles de routage *multicast* intra-domaine en deux familles : les protocoles à mode dense et les protocoles à mode épars.

Les protocoles dits à mode dense (*dense mode*) ou à forte densité de membres supposent que la bande passante est abondante, qu'un groupe *multicast* admet des

membres sur la plupart des LANs et que l'absence de membre constitue l'exception. Ces protocoles construisent un arbre basé à la source en utilisant un mécanisme d'inondation/élagage (*cf.* section 1.2.2). Un tel mécanisme est en effet d'autant moins pénalisant pour les ressources du réseau que la densité des membres du groupe est importante. Cependant, ce type de protocole n'est pas adapté aux groupes de faible densité, et en particulier aux réseaux de grande taille où les membres sont typiquement dispersés entre les différents LANs. En effet, dans ce cas on aurait souvent une transmission des paquets sur des liens ne menant à aucun destinataire. De plus, un routeur qui ne désire pas recevoir du trafic *multicast* va passer son temps à demander qu'on cesse d'en lui envoyer. Les protocoles DVMRP [RFC1075], MOSPF [RFC1247][RFC1584], et PIM-DM [ADA03] fonctionnent en mode dense.

Présentant tant de problèmes de passage à l'échelle, les protocoles à mode dense ont presque disparu au profit d'un nouveau type de protocoles dits à mode épars (*sparse mode*) ou à faible densité de membres. Ces derniers supposent, au contraire, que les membres d'un groupe *multicast* sont très dispersés et peu nombreux par rapport au nombre de LANs desservis. Afin de permettre aux sources et aux abonnés de se rencontrer sans inonder le réseau, de tels protocoles utilisent la technique d'arbre partagé. L'utilisation d'un arbre partagé permet une meilleure résistance au facteur d'échelle mais engendre une concentration du trafic au voisinage du routeur *core* et produit des chemins non optimaux (*cf.* section 1.2.3). Les protocoles CBT [RFC2189][RFC2201] et PIM-SM [RFC2362] sont à mode épars.

1.3.2.1 Le protocole DVMRP

DVMRP (*Distance Vector Multicasting Routing Protocol*) [RFC1075] est un protocole de routage *multicast* dérivé du protocole de routage *unicast* RIP (*Routing Information Protocol*) [RFC1058] et, comme lui, il utilise la notion de distance.

DVMRP construit un arbre de livraison différent pour chaque couple (source, groupe) en utilisant le mécanisme d'inondation/élagage de l'algorithme RPM (*cf.* section 1.2.2.3). DVMRP inclut son propre protocole de routage *unicast* : il construit deux tables de routages séparées pour le *multicast* et l'*unicast*. La table de routage *unicast* est construite par l'algorithme à vecteur de distance aussi connu sous le nom de *Bellman-Ford*. Cette table est utilisée par le routeur pour sélectionner la meilleure route vers la source lors de l'exécution de l'algorithme RPM.

En plus de la procédure périodique d'inondation/élagage (*cf.* section 1.2.2.3), DVMRP implémente un mécanisme d'embranchement rapide : dès qu'il reçoit un message join pour le couple (S,G), un routeur responsable de l'élagage de la branche de l'arbre (S,G) dont il était la feuille, retourne un message de greffe dit message *graft*. En recevant ce message, le routeur situé immédiatement en amont dans l'arbre de livraison annule les états installés suite à la réception du dernier message *prune*. Ce mécanisme accélère le greffage de branches précédemment élaguées.

DVRMP inclut également un procédé de *tunnelling* IP dans IP, permettant de passer des paquets *multicast* à travers des routeurs qui ne supportent pas le routage *multicast*. Les paquets *multicast* sont alors transmis encapsulés dans des paquets *uni-*

cast à travers les routeurs ne supportant pas DVMRP.

1.3.2.2 Le protocole MOSPF

MOSPF (*Multicast Extensions to OSPF*) [RFC1584], est une extension *multicast* du protocole *unicast* OSPF (*Open Shortest Path First*) [RFC1247]. Cette extension permet au protocole OSPF de router du trafic *multicast* sur un arbre basé à la source des plus courts chemins dans un domaine d'administration. MOSPF se base sur un calcul local du chemin le plus court pour toutes les sources.

OSPF est un protocole de routage *unicast* à état des liens qui divise un domaine d'administration en plusieurs zones. Chaque routeur OSPF conserve et met à jour une base de données sur les états des liens, qui décrit la topologie dans chaque zone. Cette base de données est construite grâce à la diffusion des différents états des liens par des messages LSA (*Link State Advertisement*). Chaque message LSA est diffusé par inondation à travers toute la zone de routage.

Afin d'éviter les opérations périodiques d'inondation/élagage de RPM 1.2.2.3, MOSPF introduit les messages d'appartenance aux groupes (*Group Membership LSA*) afin de permettre aux routeurs de construire l'arbre de routage *multicast*. Un routeur désigné dans une zone utilise le protocole de gestion des groupes pour s'informer de l'appartenance au groupe des membres de son sous-réseau et il est responsable de la distribution de cette information par inondation de messages Group Membership LSA vers tous les routeurs dans la zone OSPF. Alors, en utilisant la base de données de l'état des liens, chaque routeur calcule l'arbre des plus courts chemins. Par la suite, les messages *Group Membership LSA* sont utilisés pour élaguer les branches de l'arbre qui n'aboutissent pas à des membres. Pour chaque groupe *multicast*, un routeur MOSPF peut ainsi déterminer sa position sur l'arbre des plus courts chemins et mettre à jour sa table de routage *multicast*. Cette table de routage n'est pas régénérée périodiquement, mais change seulement quand il y a un changement de la topologie du réseau ou un changement de l'appartenance aux groupes. Pour ménager les ressources de tous les routeurs du réseau, le plus court chemin peut être calculé à la demande : à l'arrivée du premier paquet *multicast* pour un groupe.

1.3.2.3 Le protocole CBT

Le protocole CBT (*Core Based Trees*) [RFC2189][RFC2201] construit un arbre partagé pour chaque groupe *multicast*. Un routeur spécifique nommé *core* est assigné à chaque groupe *multicast*.

Un nouveau membre s'enregistre auprès du groupe en envoyant un message d'abonnement (IGMP ou MLD) sur son lien local. A la réception de ce message, le routeur *multicast* désigné sur le lien local envoie à destination du *core* un message d'adhésion *Join-Request*. Ce message sera intercepté et confirmé par le premier nœud faisant déjà partie de l'arbre (au pire le *core*). Un message d'acquittement *Join-Ack* sera alors envoyé, empruntant le même chemin dans le sens inverse. Ce message établit un état de routage de la forme (groupe, interface d'entrée, interface de sortie) dans les routeurs

tout le long du chemin. La figure 1.10 (a) illustre l'adhésion d'un membre à un groupe dans CBT. Le message d'adhésion dans cet exemple est intercepté par un routeur appartenant à l'arbre sur le chemin vers le *core*.

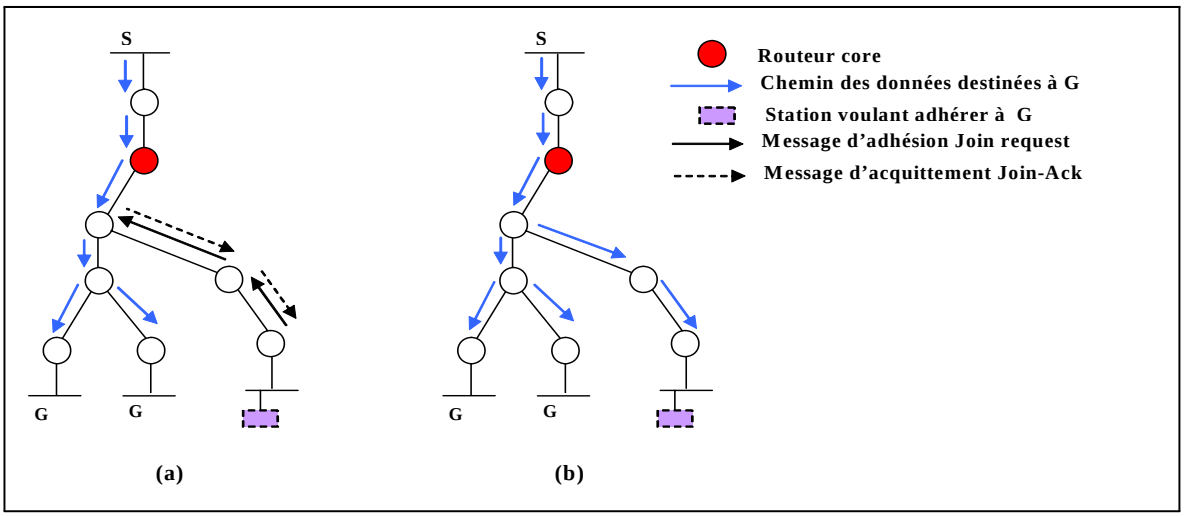


FIG. 1.10 – Le protocole CBT : procédure d'adhésion

Une branche est ainsi créée entre l'arbre existant et le routeur *multicast* ayant envoyé le message d'adhésion, permettant désormais au nouveau membre de recevoir le trafic destiné au groupe (*cf.* figure 1.10 (b)).

Un routeur qui détecte qu'il n'y a plus aucun membre appartenant à un groupe sur ses liens en aval envoie un message de notification de départ *quit-request* vers le routeur en amont. Ceci se produit lorsque la liste des interfaces filles de ce routeur devient vide pour un groupe.

Afin de maintenir la structure de l'arbre, chaque routeur envoie périodiquement un message *echo-request* indiquant à son père qu'il est toujours présent dans le groupe. Le routeur père répond avec un message *echo-reply*. Un routeur, qui ne reçoit plus de message echo request, en conclut qu'il n'y a plus de station abonnée dans le chemin des routeurs fils ; la branche est alors élaguée. Le message *echo-reply* contient la liste des groupes pour lesquels l'interface du routeur est une interface fille.

Un routeur qui ne reçoit plus des réponses *echo-request* à ses echo reply de son routeur père déduit que ce dernier n'est plus accessible à cause d'une défaillance. Les routeurs qui sont en aval auront alors à rejoindre l'arbre de nouveau. Pour ce faire, le routeur ayant détecté la défaillance transmet un message *Flush-Tree* vers toutes ses interfaces filles. Les routeurs qui reçoivent ce message retirent le (ou les) groupe(s) concerné(s) de leur table pour les interfaces concernées. Le routeur aval doit recevoir ce message sur l'interface qui correspond au bon routeur amont ; dans le cas contraire, le message est détruit. Le message reçu dans de bonnes conditions est transmis au routeur aval jusqu'aux routeurs feuilles de l'arbre. Ainsi, toutes les branches en aval du routeur ou du lien défaillant sont détruites. De nouvelles branches

seront automatiquement construites grâce au mécanisme de recensements périodiques du protocole de gestion de groupes *multicast* (IGMP ou MLD) déroulé sur les liens local des récepteurs (*cf.* section 1.3.1).

Notons enfin qu'un arbre créé par l'algorithme CBT est bidirectionnel, dans le sens que le flux de données peut être échangé dans les deux sens le long d'une branche d'arbre. Il n'y a donc pas de concept d'interfaces de sortie ou d'entrée, bien qu'il soit nécessaire de pouvoir distinguer les interfaces en amont (interfaces parentes) de celles en aval (interfaces filles). Ainsi, le trafic issu d'une source réceptrice est directement envoyé sur l'arbre *multicast* sans avoir à être d'abord encapsulé et envoyé vers le *core*.

CBT, comme tout protocole qui construit des arbres partagés, hérite à la fois des avantages et des inconvénients de ce type d'arbre.L'encapsulation des paquets *multicast* et leur envoi vers le *core* avant de les transmettre sur l'arbre engendrent un délai supplémentaire, ce qui pose des problèmes avec les applications temps réel. Le fait aussi que de nombreux groupes puissent utiliser le même routeur *core* entraîne dans certain cas une forte concentration de trafic au niveau de ce nœud central. Ceci montre l'importance de choisir le meilleur emplacement du *core* pour un groupe, voir plusieurs. Mais cela reste un problème majeur, du fait qu'on n'a pas une vue exhaustive de la topologie de l'interconnexion, ni de la composition du groupe.

1.3.2.4 Le protocole PIM

PIM (*Protocol Independent Multicast*) est un protocole de routage *multicast* indépendant du protocole de routage *unicast* sous-jacent et s'adapte donc à n'importe quel protocole *unicast* pour effectuer le routage *multicast*. Tel n'est pas le cas de DVMRP qui utilise son propre routage *unicast* et de MOSPF qui se base OSPF.

Le protocole PIM se présente sous deux formes : PIM-DM (*PIM Dense Mode*) [ADA03] et PIM-SM (*PIM Sparse Mode*) [RFC2362][FEN02]. Comme leurs noms l'indiquent, PIM-DM est un protocole à mode dense alors que PIM-SM est un protocole à mode épars. Il est à noter que les messages de contrôle PIM-DM sont intégrés à PIM-SM, de telle sorte qu'un même routeur puisse utiliser un mode différent pour chaque groupe.

1.3.2.4.1 Le protocole PIM-DM : PIM-DM [ADA03] est un protocole de routage *multicast* à mode dense assez similaire à DVMRP. Il utilise aussi la technique RPM (*cf.* section 1.2.2.3) pour construire son arbre *multicast*, avec ses deux phases qui sont l'inondation et l'élagage. Il est indépendant du protocole de routage *unicast* et s'adapte donc à n'importe quel protocole *unicast* pour effectuer le routage *multicast*. Contrairement à DVMRP qui utilise son propre routage *unicast* et à MOSPF qui utilise OSPF, PIM-DM ne construit pas sa propre table de routage *unicast*, mais accède simplement aux informations de routes indépendamment du protocole de routage *unicast* sous-jacent. Par souci de simplicité, la spécification de PIM-DM préconise de maintenir un minimum d'état, quitte à inonder un peu plus fréquemment des branches sans membres. Un routeur PIM-DM ne maintient pas une base de données des interfaces père et filles pour chaque couple (source, groupe), mais transmet un paquet sur

toutes les interfaces filles hormis celles de réception d'un message d'élagage. Un message supplémentaire de greffage permet de rétablir une arête, préalablement élaguée, lors de l'adhésion d'un nouveau membre. Notons que, comme MOSPF, PIM DM ne supporte que les arbres basés à la source.

1.3.2.4.2 Le protocole PIM-SM : Les travaux sur les algorithmes de type CBT se sont poursuivis pour venir à bout de leurs limitations persistantes tout en gardant les bonnes propriétés des arbres partagés. Ces travaux ont donné naissance à PIM-SM (*PIM Sparse-Mode*) [RFC2362][FEN02].

Contrairement à PIM-DM, PIM-SM supporte les groupes épars, dont la répartition géographique des membres, nombreux ou non, ne permet pas l'utilisation des protocoles à inondation. Il offre un support des arbres partagés, mais préconise également l'utilisation d'arbres basés à la source. Ceci lui donne un avantage face aux autres protocoles basés seulement sur la technique d'arbre partagé comme le protocole CBT.

Les décisions de routage sont basées, comme pour PIM-DM, sur l'existence d'une table de routage *unicast* quelconque et la construction de l'arbre est indépendante d'un protocole de routage *unicast* particulier.

L'équivalent du *core*, le routeur autour duquel est construit un arbre CBT est appelé un point de rendez-vous (*Rendez-vous Point*) ou RP dans PIM. Malgré une dénomination différente, le rôle d'un RP est quasiment identique à celui d'un *core*. Lorsqu'un récepteur s'abonne au groupe, son routeur désigné local (DR) génère un message *join* directement adressé au RP. Le message join est alors traité par chacun des routeurs qui séparent le futur abonné du RP en marquant l'interface par laquelle le message join est reçu. Les états maintenus au niveau de ces routeurs sont de la forme (*,G). La figure 1.11 illustre l'adhésion d'un hôte à un groupe *multicast* dans PIM-SM.

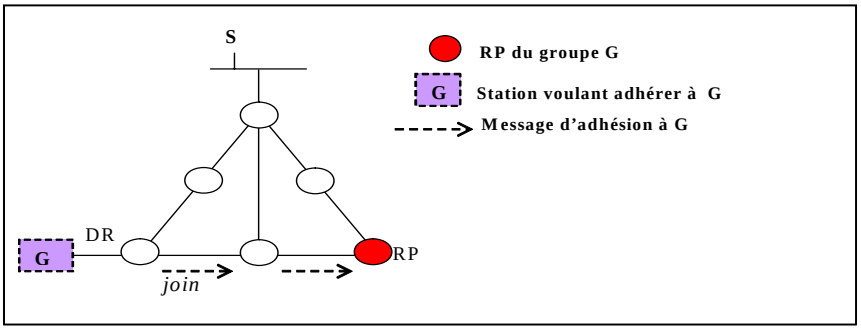


FIG. 1.11 – Le protocole PIM-SM : procédure d'adhésion

Lorsqu'une source commence à transmettre des paquets vers un groupe donné, le DR de cette source encapsule les premiers paquets dans des paquets *unicast* appelés paquets d'enregistrement et les transmet vers le RP de ce groupe en mode *unicast*. Après la réception de ces paquets d'enregistrement, le RP envoie un message d'adhésion vers le DR de cette source. Chaque routeur intermédiaire sur le chemin du RP vers le DR de la source ajoute alors une nouvelle entrée spécifique à la source S et au groupe G et notée (S,G) dans sa table de routage *multicast*. Ainsi, les paquets *multicast* seront transmis nativement (sans encapsulation) vers le RP qui les transmet à son tour vers les destinataires membres du groupe. Il faut noter que jusqu'au moment où ces entrées sont ajoutées dans les tables de routage intermédiaires, tous les paquets *multicast* sont transmis comme des paquets *unicast* encapsulés (*cf.* figure 1.12).

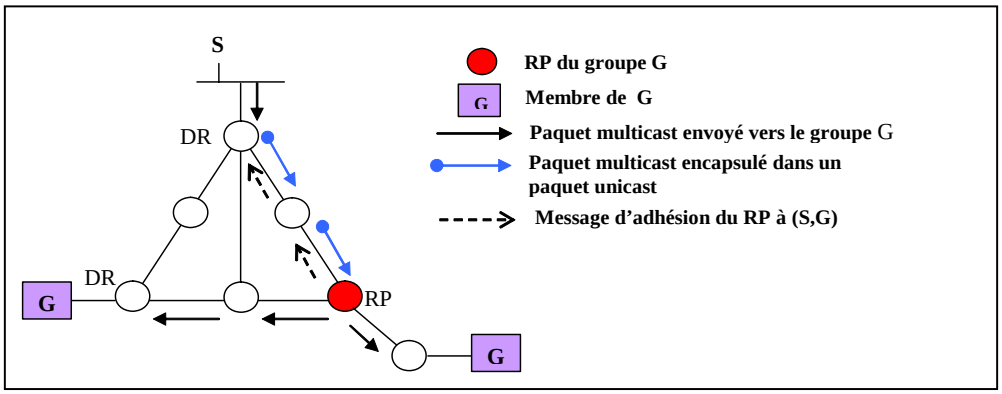


FIG. 1.12 – Le protocole PIM-SM : transmission des paquets de données

Les arbres partagés unidirectionnels résultant de l'exécution de PIM-SM ne sont pas forcément les plus efficaces. Ils constituent cependant le seul moyen d'indiquer aux récepteurs que des données commencent à arriver. Sur réception des premières données, le routeur local d'un récepteur peut décider de changer d'arbre en initiant la construction d'un arbre des plus courts chemins (SPT : *Short Path Tree*). Pour cela, le routeur envoie un message *join* directement adressé à la source. Dès que les données commencent à arriver sur l'arbre SPT nouvellement construit, un message *prune* peut alors être retourné au point de rendez-vous pour éviter de recevoir les données diffusées en double. La procédure de basculement entre arbre partagé et arbre basé à la source est illustrée par la figure 1.13.

Contrairement aux autres protocoles qui construisent des arbres de coût minimal tels que DVMRP et PIM-DM, les informations de routage créées par PIM-SM sont maintenues uniquement le long des arbres SPT. Dans DVMRP et PIM-DM, les informations d'élagage sont maintenues le long des chemins à l'extrémité desquels il n'y a pas de récepteur abonné à la session en cours.

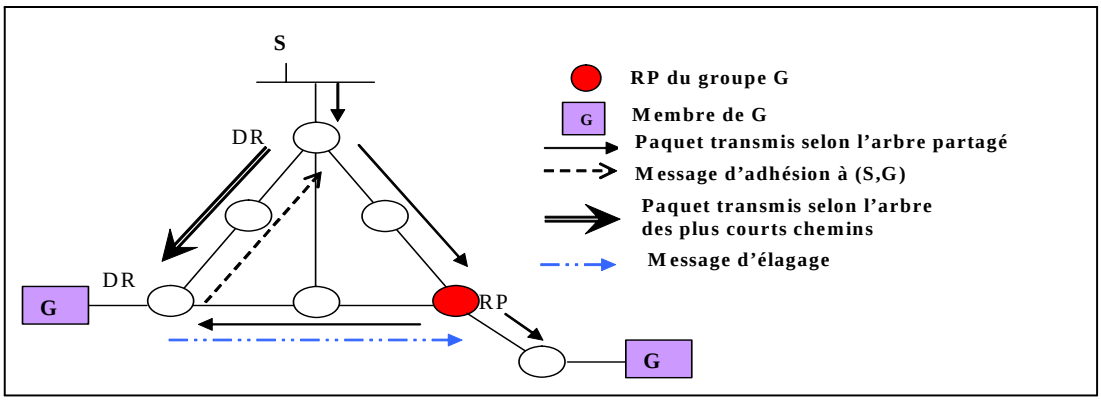


FIG. 1.13 – Le protocole PIM-SM : basculement vers l'arbre des plus courts chemins

1.3.3 Les protocoles de routage *multicast* inter-domaines

Les domaines doivent pouvoir s'échanger entre eux le trafic *multicast* pour permettre à tout hôte de l'Internet de recevoir du trafic *multicast* provenant d'une source n'appartenant pas à son domaine. Il a été décidé d'avoir un RP par domaine car les fournisseurs de service Internet (ISP) ne veulent pas transporter un trafic pour lequel ils n'ont ni récepteur ni émetteur. Ceci a créé la nécessité d'avoir une coordination inter-domaines entre RPs, ce qui peut être réalisé par le protocole MSDP (*Multicast Source Discovery Protocol*) [RFC3618]. Le but de MSDP est de permettre aux RPs de découvrir les sources appartenant à d'autres systèmes autonomes. Si les sources doivent transmettre du trafic *multicast* à des récepteurs appartenant à un autre domaine, un arbre partagé est construit en utilisant PIM-SM. Les paquets *multicast* seront acheminés au-dessus de cet arbre de distribution inter-domaines.

Comme il existe des routeurs qui ne sont pas doués du *multicast*, la topologie *multicast* peut différer de la topologie *unicast*. Pour pouvoir gérer ceci, une extension a été introduite au protocole BGP (*Border Gateway Protocol*) [RFC1771] donnant lieu au protocole MBGP (*Multiprotocol BGP*) [RFC2283]. Le protocole MBGP permet de définir des politiques de routage *unicast* et *multicast* séparées.

Il est cependant à noter que le *multicast* inter-domaines est fondamentalement différent pour IPv4 et IPv6. En effet, personne n'a voulu déployer MSDP (*Multicast Source Discovery Protocol*), protocole lourd et compliqué, déjà frein du *multicast* IPv4. La communauté IETF s'oriente vers le modèle SSM (*Source Specific Multicast*), dans lequel les récepteurs spécifient les adresses des sources (*cf.* section 1.4). Si le modèle SSM est très adapté aux diffusions de télévision ou de radio sur Internet, où les sources sont connues à l'avance ; il ne l'est pas pour les groupes où il y a plusieurs émetteurs non connus à l'avance, et pouvant varier pendant les sessions *multicast*. Des travaux sont en cours pour permettre la découverte des sources dans le modèle SSM. Le passage à SSM ne fait cependant pas l'unanimité. Cela nécessite le support de MLDv2 sur tous les systèmes d'exploitation, ainsi que sur les applications *multicast*. Si des progrès sont réalisés dans ce domaine, des solutions sont nécessaires à court

terme pour permettre l'interdomaine en ASM (*Any Source Multicast*), ou *multicast* "classique" dans lequel une application s'abonne à un groupe et reçoit le trafic de toutes les sources du groupe. La solution retenue, appelée *Embedded-RP* [RFC3956] consiste à insérer l'adresse du RP à l'intérieur de l'adresse *multicast*. Cette approche change complètement le modèle connu à ce jour puisque le RP (Point de Rendez-vous) n'est plus forcément local. La notion de domaines *multicast* telle qu'on les connaît aujourd'hui disparaît. Le réseau devient un unique domaine *multicast*, dans lequel des RP sont configurés et sont partagés.

1.4 Evolution du modèle de *multicast* IP

De nos jours, il existe deux modèles de *multicast* IP normalisés par l'IETF : le modèle ASM (*Any Source Multicast*) et le modèle SSM (*Source Specific Multicast*) [HOL03][RFC3569], basés tous les deux sur le modèle Hôte Groupe.

Le modèle ASM est le *multicast* originel de Deering [DEE91], qui a été renommé ainsi lorsque d'autres modèles de *multicast* sont apparus. Il est aussi parfois appelé le *multicast* d'Internet Standard (ISM) ou le *multicast* Indépendant de la Source (SIM). Il fournit un service *multicast* de plusieurs à plusieurs. Un membre qui s'abonne à un groupe reçoit les flux *multicast* provenant de toutes les sources qui envoient à ce groupe (*cf.* figure 1.14). Le modèle ASM est particulièrement adapté à la vidéoconférence.

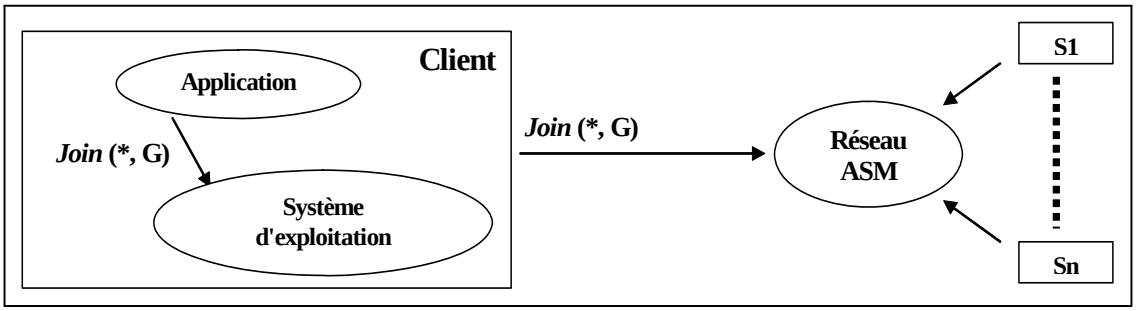


FIG. 1.14 – Le modèle ASM du *multicast* IP

Le modèle ASM présente un problème de contrôle d'accès lors de l'envoi de données vers un groupe, puisque rien n'empêche quiconque d'envoyer des données vers un groupe. De plus, les mécanismes garantissant l'unicité des adresses *multicast* dans l'Internet se sont révélés trop compliqués par rapport au but recherché.

Le modèle SSM a été récemment proposé pour remédier aux problèmes liés à la multitude de sources dans le modèle ASM. Ce modèle ne permet la diffusion qu'à partir d'une source vers plusieurs destinataires ("de un à plusieurs"). Un membre s'abonne à un couple (S, G), où S désigne l'adresse de la source et G celle du groupe. Ce couple est appelé canal *multicast* (*cf.* figure 1.15).

Le modèle SSM permet le contrôle de la transmission des flux *multicast*. De plus, il résout le problème d'allocation d'adresses *multicast* puisque l'unicité de telles adresses

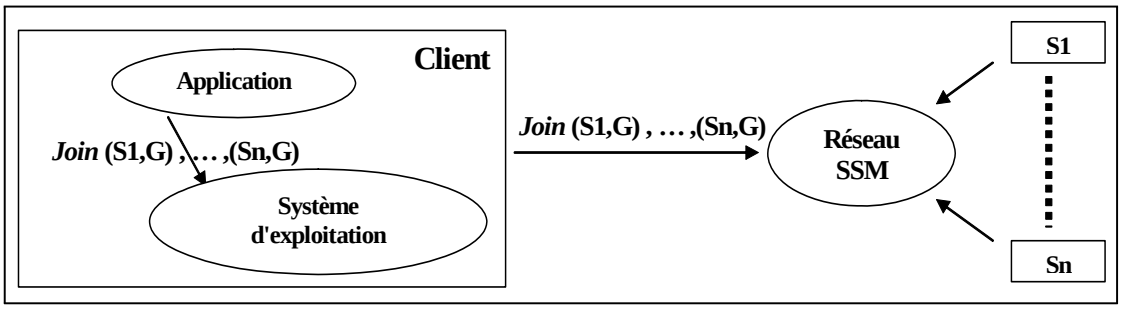


FIG. 1.15 – Le modèle SSM du *multicast* IP

n'est plus assurée globalement mais au niveau de chaque source. Ce modèle est plus particulièrement destiné à la diffusion en direct (par exemple de radios ou de télévisions sur Internet).

1.5 Conclusion

Le concept de communication de groupe favorise la mise en correspondance d'un ensemble de stations. Le *multicast* IP a été proposé pour éviter la transmission multiple d'un même paquet dans les communications de groupe. Cependant, le *multicast* IP engendre des coûts importants reliés d'une part au modèle Hôte Groupe, et, d'autre part, aux protocoles de routage *multicast* mis en jeu.

Dans le modèle Hôte groupe de Deering [DEE91], un groupe est identifié par une adresse *multicast* employée pour la gestion des abonnements et pour le routage. Cette adresse doit être unique dans sa portée qui est souvent globale. L'allocation d'adresse présente un problème majeur pour les protocoles de routage *multicast* interdomaines, vu le risque d'avoir une collision d'adresses entre différentes applications. Les solutions proposées pour résoudre ce problème ne répondent pas aux exigences des applications vu la nature dynamique de l'allocation d'adresse *multicast*. L'anonymat des destinations est un autre problème lié au modèle Hôte Groupe. Du fait que les routeurs intermédiaires acheminent les paquets *multicast* sans connaître ni les destinations finales, ni le nombre de fois que ces paquets seront dupliqués plus tard dans le réseau, la gestion de la sécurité, de la facturation et de la politique de routage devient compliquée.

En plus des coûts du modèle Hôte Groupe, les protocoles de routage *multicast* engendrent une signalisation importante et des états de routages par groupe ou canal *multicast* au niveau des routeurs intermédiaires. Afin de maintenir l'état des arbres de livraison *multicast*, les protocoles de routage *multicast* effectuent des échanges de messages de contrôle afin de créer, pour chaque arbre, un état dans chaque routeur concerné. Ceci peut engendrer des tables de routage *multicast* énormes. Les protocoles de routage *multicast* à mode dense diffusent même cette information de routage aux endroits où elle n'est pas nécessaire. Les protocoles à mode épars essayent de limiter

la quantité d'information de routage *multicast* devant être diffusée, traitée et stockée dans tout le réseau. De tels protocoles (par exemple CBT) emploient un arbre de livraison partagé par toutes les sources envoyant vers un groupe *multicast* et essayent de limiter la distribution d'information de routage *multicast* aux nœuds qui en ont vraiment besoin. Cependant, l'utilisation de l'arbre partagé affecte l'optimalité des chemins et crée une concentration du trafic dans de petites zones du réseau. De plus, la signalisation et le nombre d'états par flux créés pas ces schémas restent toujours importants. Outre la construction et le maintien de l'arbre de livraison *multicast*, un mécanisme d'annonce de la source est indispensable pour permettre aux membres de "se relier" aux sources d'un groupe donné sans connaître les sources elles-mêmes. Dans les protocoles à mode épars, ceci est réalisé en ayant un point de rendez-vous, qui doit être annoncé dans le domaine complet. Pour les protocoles à mode dense, ceci est réalisé par un mécanisme d'inondation/élagage. Les deux approches présentent des problèmes additionnels de passage à l'échelle. De plus, les protocoles de routage inter-domaines génèrent un échange d'informations de routages entre les différents domaines hébergeant des sources ou des membres *multicast*.

Le *multicast* à Source Spécifique (SSM) remédie à certains de ces problèmes : un nœud rejoint une source spécifique du groupe, ainsi le canal est identifié par le couple (S, G), ou S désigne l'adresse de la source et G désigne celle du groupe *multicast*. Cette approche permet d'éviter l'attribution d'adresse *multicast* à échelle globale. De plus, elle rend non nécessaire l'utilisation d'un protocole de routage inter-domaine. L'annonce de la source ne s'effectue plus au niveau du protocole de routage, elle devient hors bande (par exemple à travers une page Web). Mais il est à noter que même avec SSM, il y a toujours création de l'état et de la signalisation par canal *multicast* dans chaque nœud de l'arbre.

Chapitre 2

Le *Multi-unicast* Explicite

Les schémas traditionnels du *multicast* IP ont des coûts importants liés essentiellement à l'attribution d'adresses *multicast*, à l'anonymat des destinations, à la mémorisation des états des arbres au niveau des routeurs et aux échanges de l'information de routage. De plus, bien qu'ils permettent de supporter des groupes de très grande taille, ils présentent des problèmes de passage à l'échelle quand le nombre de groupes *multicast* dans le réseau augmente.

Le *multicast* à Source Spécifique (SSM) [HOL03][RFC3569] permet d'éviter l'attribution d'adresse *multicast* à échelle globale. De plus, il rend non nécessaire l'utilisation d'un protocole de routage inter-domaine. Cependant, SSM augmente la taille des états de routage et de la signalisation générés puisque il utilise la technique d'arbre basé à la source.

Il est aussi à noter que le coût de maintien de l'arbre *multicast* est d'autant plus important que la taille des groupes est petite et que les membres sont dispersés, puisque, dans ce cas, il est mal compensé par le gain du à la transmission en *multicast*. Or les petits groupes sont typiques pour la communication, le jeu et les applications de collaboration. De plus, Leur nombre sur Internet est de plus en plus important. Ceci a mené à la recherche de nouveaux schémas de transmission multipoint qui s'adaptent mieux aux petits groupes et qui passent à l'échelle quand le nombre de groupes *multicast* devient énorme.

Dans ce cadre, un nouveau protocole de transmission multipoint appelé le *multi-unicast* explicite ou Xcast (*Explicit Multi-unicast*) [BOI05] a été proposé. A la différence des protocoles de routage *multicast* qui utilisent une adresse de groupe, Xcast est basé sur l'encodage explicite de la liste des adresses *unicast* des différentes destinations dans les paquets de données, permettant ainsi l'utilisation des tables de routage *unicast* traditionnelles. Ceci permet d'éliminer la signalisation et l'information d'état par session, et par conséquent, de tolérer la coexistence d'un nombre illimité de sessions *multicast* dans un réseau. En contrepartie, l'encodage explicite des adresses de toutes les destinations dans les paquets de données impose une limitation à la taille des groupes *multicast*.

Les protocoles Xcast+ [MYU01] et GXcast [BOU03] sont des extensions du protocole Xcast qui ont été proposées dans le but d'améliorer les performance de Xcast.

En premier lieu, nous décrivons le protocole Xcast, ses avantages, ses inconvénients et son domaine d'applicabilité. Nous présentons ensuite les extensions apportées par les protocoles Xcast+ et GXcast à la spécification de base de Xcast.

2.1 Le protocole Xcast

2.1.1 Description

Dans le modèle Hôte Groupe, un paquet de données porte une adresse *multicast* comme marque logique de tous les membres du groupe. Dans Xcast, la source maintient les adresses *unicast* des différents destinataires dans le canal auquel elle veut transmettre un flux *multicast*. Elle encode la liste de ces adresses dans l'en-tête Xcast de chaque paquet sortant. Tout au long du chemin, chaque routeur intermédiaire analyse l'en-tête du paquet Xcast reçu, partitionne les destinations selon le prochain nœud à atteindre, puis envoie à chaque prochain nœud un paquet avec un en-tête Xcast approprié.

Quand il reste une seule destination, le paquet Xcast peut être converti en un paquet *unicast* ordinaire qui sera envoyé le long du chemin qui reste. Ce mécanisme est appelé le X2U (*Xcast To unicast*).

La liste de destinations est encodée dans un en-tête séparé. L'en-tête Xcast pour IPv4 (Xcast4) est placé entre l'en-tête IPv4 et l'en-tête de la couche transport, le paquet aura alors la forme suivante :

[En-tête IPv4 | En-tête Xcast4 | En-tête de transport | Charge utile]

Le codage de l'en-tête Xcast sur IPv6 (Xcast6) est semblable à celle sur IPv4, sauf qu'elle est contenue dans les en-têtes d'extension d'IPv6 :

[En-tête IPv6 | En-tête d'extension Xcast6 | En-tête de transport | Charge utile]

L'adresse source de l'en-tête IP est celle de la source de la session Xcast, et l'adresse destination est l'adresse *multicast All_Xcast_Routers*. Le format de l'en-tête Xcast6 est détaillé dans l'annexe B.

Le traitement qu'un routeur effectue à la réception d'un paquet Xcast est décrit par l'algorithme suivant :

1. pour chacune des destinations énumérées dans le paquet, consulter la table de routage *unicast* pour déterminer le prochain nœud correspondant.
2. partitionner l'ensemble de destinations selon leurs prochains noeuds.
3. dupliquer le paquet de sorte qu'il y ait une copie pour chacun des prochains noeuds trouvés dans les étapes précédentes.
4. modifier la liste de destinations dans chacune des copies de sorte que la liste dans la copie relative à un prochain noeud donné inclue la liste des destinations admettant ce nœud comme prochain nœud dans la table de routage.

5. envoyer les copies modifiées du paquet aux prochains noeuds.
6. optimisation : S'il y a seulement une destination pour un prochain noeud donné, le paquet peut être envoyé comme paquet *unicast* standard à la destination (X2U).

2.1.2 Exemple

Considérons l'exemple représenté sur la figure 2.1.

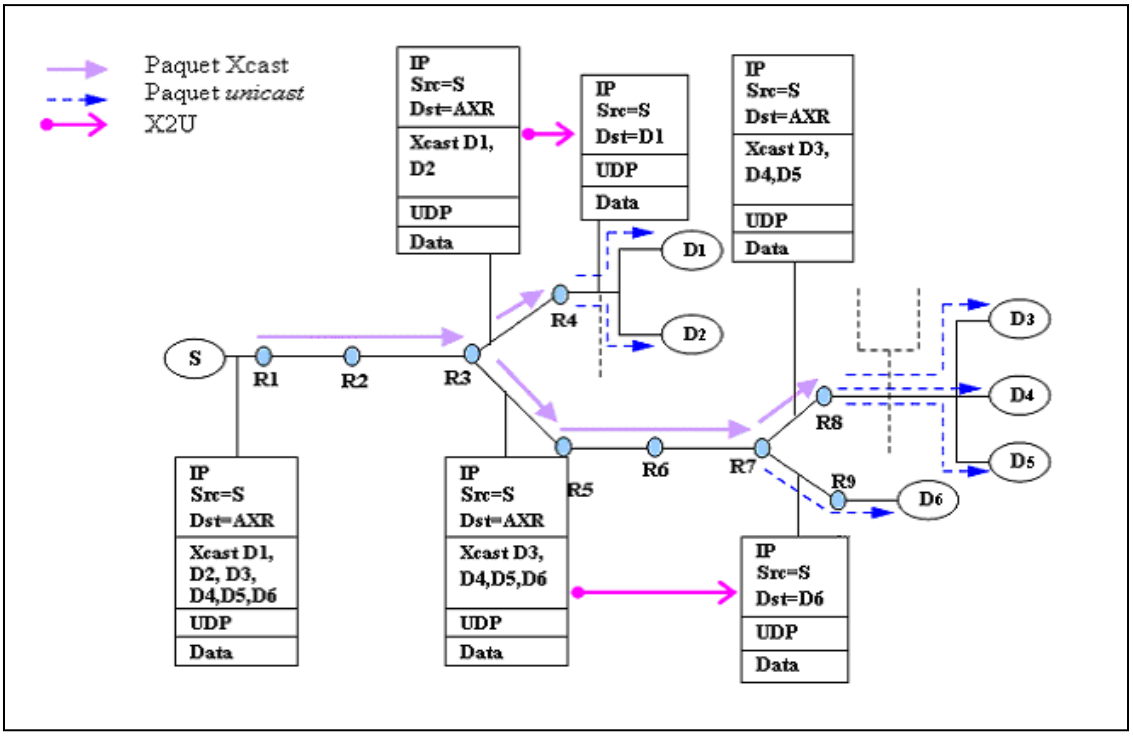


FIG. 2.1 – Exemple de transmission d'un paquet Xcast

Supposons que la source S émet un paquet Xcast contenant la liste des destinations (D1,D2,D3,D4,D5,D6). Lorsque R1 reçoit ce paquet, il remarque que R2 est le prochain nœud sur le chemin vers toutes les destinations D1 à D6, il transmet alors le paquet tel qu'il est à R2. De même, R2 transmet le paquet inchangé à R3. Ce dernier partitionne alors les destinataires selon le prochain nœud : il transmet à R4 une copie destinée à (D1,D2), et à R5 une autre copie destinée à (D3,D4,D5,D6). Comme il ne reste qu'un saut respectivement vers D1 et D2, R4 leur envoie le paquet sous forme *unicast* standard (X2U). Le paquet est acheminé selon le même principe sur la branche allant de R3 au reste des destinataires. Notons que R7 transmet une copie Xcast de destination (D3,D4,D5) à R8 et une copie *unicast* de destination D6 à R9.

2.1.3 Avantages et inconvénients

Xcast tire profit de l'un des principes fondamentaux de la philosophie d'Internet, à savoir le déplacement de la complexité du cœur du réseau vers ses bords. Ce principe a guidé la conception d'IP et de TCP et a rendu possible la croissance incroyable d'Internet. A titre d'exemple, les routeurs du cœur d'Internet manipulent de grands blocs CIDR et non pas des adresses ou des connexions individuelles, chose qui a permis l'expansion importante de ce réseau en nombre de hôtes. De même, Xcast est basée sur l'idée que les routeurs ne devraient pas maintenir un grand nombre de différents flux *multicast*.

Comparé au *multicast* traditionnel, Xcast présente les avantages suivants :

1. les routeurs n'ont pas à mettre à jour l'état par session (ou par canal). Ceci rend Xcast très scalable en terme de nombre de sessions pouvant être supportées. En effet, les nœuds dans le réseau n'ont besoin ni de diffuser ni de stocker des informations de routage *multicast* pour ces sessions.
2. aucune allocation d'adresse *multicast* n'est exigée.
3. aucun protocole de routage *multicast* intra ou inter-domaine n'est nécessaire pour acheminer les paquets Xcast. En effet, ceux-ci prennent toujours le chemin "droit" déterminé par les protocoles ordinaires de routage *unicast*. Ceci veut dire aussi qu'on n'a plus besoin d'autres tables de routage que la table *unicast*.
4. pas de nœud de vulnérabilité unique, ce qui augmente la robustesse du réseau face aux pannes.
5. même dans le cas de chemins asymétriques, le routage est optimal, contrairement aux protocoles de routage *multicast* traditionnels qui créent des arbres non optimaux si les chemins sont asymétriques . Or il est attendu que le nombre de chemins asymétriques dans l'Internet augmente à cause des politiques de routage et de l'ingénierie du trafic. De tels schémas mèneront alors à un taux croissant de routage non optimal.
6. la réaction de Xcast aux changements de chemins *unicast* est automatique et immédiate. Pour les protocoles de routage *multicast* traditionnels, un échange d'informations entre les protocoles de routage *unicast* et *multicast* est nécessaire. Ceci est réalisé en général sur la base d'interrogation, ce qui mène à une réaction plus lente (par exemple en cas de rupture d'un lien). De plus, le temps de rétablissement augmente avec le nombre de groupes pour de tels schémas.
7. la gestion de la sécurité et de la facturation est plus facile puisque les sources dans Xcast connaissent les membres du canal, et peuvent par conséquent rejeter certains membres ou compter le trafic sortant par membre. Un routeur de bordure peut également déterminer le nombre de fois qu'un paquet sera dupliqué dans son domaine, ce qui lui permet par exemple de limiter la bande passante utilisée par émetteur.
8. à côté de la liste des destinations, un paquet Xcast peut contenir une liste de *DiffServ CodePoints* (DSCPs), ce qui permet d'avoir des récepteurs avec dif-

férentes classes de service dans le même canal, alors que les protocoles *multicast* traditionnels doivent créer un groupe séparé pour chaque classe de service.

9. Xcast peut tirer profit de l'ingénierie de trafic appliquée aux chemins *unicast*.
10. la mise en place de protocoles fiables au dessus Xcast est simple. En effet, Xcast peut facilement adresser un sous-ensemble de la liste des destinations initiale pour faire une retransmission.

L'inconvénient principal de Xcast est l'introduction d'une complexité importante sur le traitement de l'en-tête des paquets par les routeurs intermédiaires. En effet, ces derniers doivent effectuer une consultation de la table de routage par destination. De plus, ils doivent gérer la duplication des paquets et la création d'un en-tête pour chaque prochain nœud.

Cependant, il est à noter que :

- Puisque Xcast sera typiquement employé pour des sessions super dispersées, le nombre de points de branchement sera assez réduit comparé à celui de points de non branchement. Or il n'y a construction de nouveaux en-têtes qu'au niveau des points de branchement.
- Au niveau de plusieurs points de non branchement, une seule destination reste, donc le paquet pourra être envoyé en *unicast* (X2U).
- Le routage peut être accéléré en employant un codage hiérarchique de la liste des destinations en combinaison avec l'agrégation dans les tables de routage [BOI05].
- Quand le paquet arrive à une région du réseau où la conservation de la bande passante n'est plus un souci, il peut être transformé par un X2U prématuré. Le X2U prématuré se produit quand un routeur décide de transformer le paquet Xcast en des paquets *unicast*, plus simples à traiter dans le reste du chemin.

Par ailleurs, un problème majeur du protocole Xcast est qu'il ne supporte que des groupes de taille réduite, à cause de la limitation imposée à la taille d'un paquet IP.

2.1.4 Applicabilité

Il semble y avoir deux types de *multicast* qui sont assez répandus : un *multicast* qui envoie des données à un très grand nombre de destinations (*broadcast-like multicast*) et un *multicast* qui envoie des données à un groupe de taille réduite (*narrowcast*). Un exemple du premier type est la multi-diffusion audio visuelle d'une présentation à tous les employés dans un Intranet d'une entreprise. Un exemple du second est une vidéoconférence impliquant 3 ou 4 participants. Pour les raisons de passage à l'échelle, il semble prudent d'employer différents mécanismes dans les deux cas. En effet, un protocole unique est incapable de répondre aux exigences des différentes applications.

Xcast n'est pas approprié aux sessions multipoint à grand nombre de membres, tels que l'émission d'une réunion de l'IETF. Cependant, il fournit un complémentaire important aux schémas *multicast* existants, puisqu'il peut supporter un très grand nombre de petites sessions. Ainsi, il est adapté à des applications importantes telles que la téléphonie sur IP, la vidéoconférence, les jeux sur Internet, les applications de

collaboration ... etc, pour lesquelles il y a typiquement un grand nombre de groupes de petite taille.

Jusqu'à présent, l'envoi en *unicast* est généralement utilisé pour ce type d'applications. Il peut même paraître inutile d'utiliser le *multicast* pour les groupes à nombre réduit de membres. Cependant, le *multicast* prend extrêmement de l'importance pour résoudre le problème de la bande passante limitée en "dernier mille", comme l'illustre l'exemple suivant : soient n utilisateurs réunis dans une vidéoconférence. En général, les technologies d'accès sont asymétriques (par exemple xDSL, GPRS ou câble modem). Ainsi, un hôte avec une connexion xDSL n'à aucun problème pour recevoir les n-1 canaux vidéo à 100kb/s, mais il ne peut pas envoyer ses propres signaux vidéo n-1 fois à ce débit. Ainsi, en raison de la capacité d'accès limitée et souvent asymétrique, une certaine forme de *multicast* est exigée. Ainsi, une application simple mais importante de Xcast serait la formation d'une sorte de pont vers le lien d'accès : le hôte envoie le paquet Xcast avec la liste d'adresses *unicast* et le premier routeur exécute un X2U prématuré.

2.2 Le protocole Xcast+

Le protocole Xcast+ est une extension de Xcast qui a été proposé par *Myung-Ki et al* dans [MYU01], d'une part pour résoudre le problème de passage à l'échelle de Xcast dans le cas d'un groupe de taille moyenne, et d'autre part, pour fournir un plan de contrôle permettant de supporter les sources et les récepteurs *multicast* IP standard. Ceci est assuré en utilisant les protocoles de gestion des groupes *multicast* IGMPv3 [RFC3376] et MLDv2 [RFC3810] respectivement dans IPv4 et IPv6.

La version IPv6 de Xcast+ est appelée Xcast+6, elle se base sur le protocole MLDv2 pour la gestion des abonnements au niveau lien local. Dans Xcast+6, chaque lien local dispose d'un routeur désigné (DR). Un récepteur désirant faire partie d'une session *multicast* identifiée par le canal (S,G) émet un rapport d'abonnement MLDv2 spécifique à (S,G) (*cf.* chapitre 1 section 1.3.1). Quand le DR associé à ce récepteur reçoit ce rapport, il envoie à la source S une demande d'enregistrement Xcast+ contenant l'adresse de la source S, l'adresse de groupe G, et sa propre adresse. Le DR associé à la source maintient la liste des adresses de tous les DR des récepteurs appartenant au canal (S,G). Il intercepte la demande d'enregistrement et ajoute à cette liste l'adresse du DR du récepteur ayant demandé l'adhésion.

Lorsque la source envoie un paquet *multicast* vers (S,G), son DR intercepte ce paquet et crée un paquet Xcast+ dans lequel il encode explicitement la liste des DR de destinations enregistrés auprès de lui à (S,G), il complète le paquet avec les données à envoyer et l'émet vers le(s) prochain(s) routeur(s) concerné(s) (M2X : *multicast to Xcast+*). Le chemin suivi par le paquet Xcast+ est le même que celui que suivrait un paquet Xcast. Une autre différence avec Xcast a lieu au niveau des DR récepteurs : les paquets Xcast+ qui leur parviennent sont convertis en paquets *multicast* et envoyés aux réseaux dont les DR se chargent (X2M : *Xcast+ to multicast*).

2.3 Le protocole GXcast

Un paquet IP peut être fragmenté par un nœud intermédiaire sur son chemin si sa taille dépasse la capacité du lien qu'il doit emprunter appelée la MTU (*Maximum Transfert Unit*) du lien. La fragmentation IP est un mécanisme qui tronque un paquet IP en plusieurs paquets IP autonomes (c'est-à-dire muni chacun d'un en-tête IP valide) et qui partage les données entre ces paquets.

Or un paquet Xcast est encapsulé dans un paquet IP, qui risque alors de devenir de taille importante si la liste des destinations est longue, ce qui peut provoquer une fragmentation IP au niveau d'un lien dont la MTU est inférieure à la taille du paquet. Par ailleurs, bien que le protocole Xcast+ réduit considérablement la taille de la liste des adresses *unicast* encodées dans le paquet, il n'élimine pas totalement le risque de fragmentation IP puisque le nombre de DRs de destinations peut devenir important. La figure 2.2 montre l'effet qu'aurait la fragmentation IP sur un paquet Xcast (ou Xcast+).

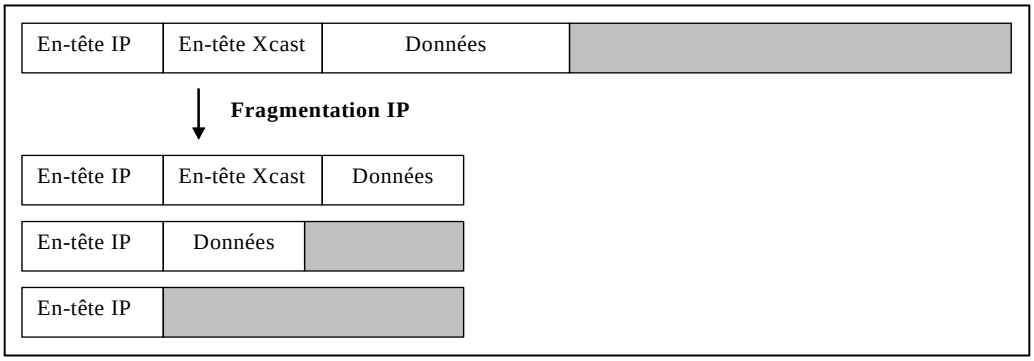


FIG. 2.2 – Effet de la fragmentation IP sur un paquet Xcast

Nous pouvons remarquer que seul le premier paquet résultant est un paquet Xcast valide, puisqu'il est le seul paquet à contenir un en-tête Xcast (il ne comporte cependant aucune données provenant de l'application). Les deux autres paquets ne seront pas traités comme des paquets Xcast : ils ne parviendront pas aux destinataires. Pour interdire à un paquet Xcast d'être fragmenté, le drapeau DF (*Don't Fragment*) de son en-tête IP doit être positionné à 1. Dans ce cas, si un lien est incapable de transmettre le paquet sans le fragmenter, le paquet est rejeté.

Si IPv4 est utilisé, une source qui veut garantir la livraison des paquet Xcast aux destinataires peut limiter la taille d'un paquet à 576 octets qui correspond à la MTU minimale garantie par IPv4 sur un lien. Cette taille limite le nombre de destinataires pour une session Xcast à 135. Cependant, le nombre de destinataires d'une session Xcast dépend d'un champ de 7 bits, ce qui limite la taille d'une session à 127 destinataires. Pour IPv6, puisque la MTU minimale garantie est de 1280 octets et que la taille d'une adresse IPv6 est de 16 octets, la taille d'un groupe Xcast est limitée à 74 destinataires.

Pour remédier à ces restrictions sur la taille d'un groupe Xcast, le protocole GXcast (*Generalized Xcast*) a été proposé [BOU03]. GXcast est une généralisation de la technique Xcast conçue pour résoudre le problème de la fragmentation et pour supporter un plus grand nombre de membres par groupe. Il utilise le même plan de contrôle que le protocole Xcast+ mais limite explicitement, au niveau de la source, le nombre maximal nm de destinataires encodés dans l'en-tête d'un paquet. La source dans GXcast partitionne la liste initiale L de destinataires en plusieurs sous-listes de destinataires Li. Chacune de ces listes Li contiendra au plus nm membres. Autant de paquets Gxcast que de listes Li seront envoyés, chacun contenant dans l'en-tête GXcast les destinations contenues dans la liste correspondante. Il s'agit donc d'un mécanisme de fragmentation à la source qui a pour but de prévenir une fragmentation IP éventuelle au sein du réseau.

2.4 Conclusion

Le routage *multi-unicast* explicite offre une alternative intéressante quand il s'agit de gérer un très grand nombre de sessions multipoint de petite taille. Cependant, ne pouvant pas supporter les groupes à grand nombre de membres, les techniques Xcast ne sont pas conçues pour remplacer le modèle traditionnel du *multicast* IP mais pour le complémenter.

La spécification de base du routage *multi-unicast* explicite est définie par le protocole Xcast [BOI05]. Or ce protocole spécifie uniquement le plan de données et laisse le contrôle des sessions multipoint à la charge des couches supérieures. De plus, l'encodage de la liste de toutes les destinations finales dans l'en-tête Xcast réduit considérablement la taille maximale toléré d'un groupe Xcast. Pour pallier à ces problèmes, le protocole Xcast+ [MYU01] étend le protocole Xcast par un plan de contrôle basé sur l'utilisation des protocoles de gestion des abonnements *multicast* traditionnels (IGMPv3 pour IPv4 et MLDv2 pour IPv6), ce qui permet de supporter les hôtes *multicast* standard. De plus, l'encodage d'une liste de routeurs désignés au lieu de la liste exhaustive des destinations finales permet à Xcast+ de gérer les groupes de taille moyenne.

Une partie des inconvénients des protocoles Xcast et Xcast+ vient de leur incapacité à gérer la fragmentation des paquets. De plus, une limite forte existe sur le nombre de membres du groupe *multicast*. Une généralisation de ces protocoles, nommée GXcast [BOU03], a été alors proposée. GXcast permet de résoudre le problème de la fragmentation en utilisant un mécanisme de fragmentation à la source.

Chapitre 3

Mobilité des Réseaux dans IPV6

Le protocole IP est le standard de communication réseau dans le monde d'Internet. La majorité des applications existantes sont développées et conçues pour utiliser ce protocole. En effet, celui-ci offre à une machine la possibilité d'être à la fois reliée à son réseau local, et de dialoguer avec n'importe quelle autre machine sur l'Internet.

Par ailleurs, l'apparition, grâce à la miniaturisation, d'équipements informatiques de plus en plus petits et puissants, combinée à l'émergence des technologies de transmission sans fil, a favorisé le développement et l'amplification des nouveaux services de mobilité. Cependant, le modèle TCP/IP ayant été conçu bien avant l'apparition des nouvelles technologies de la mobilité, ses fonctionnalités ne peuvent tenir compte de ce nouveau type de machines. Afin de supporter les nœuds mobiles dans l'Internet, l'IETF a proposé le protocole Mobile IP, qui est une extension de IP. Les deux versions de Mobile IP, à savoir Mobile IPv4 [RFC3344] et Mobile IPv6 [RFC3775] constituent le standard IETF pour la gestion de la mobilité respectivement dans les réseaux IPv4 et IPv6.

Les travaux dans le domaine de la mobilité dans l'Internet ont été jusqu'à récemment consacrés au support des stations mobiles, dans le but de fournir une connectivité Internet permanente à ces stations, et de maintenir leurs communications en cours alors qu'ils sont entrain de se déplacer. Or, il est possible qu'un réseau tout entier en fasse autant. Il s'agit alors d'un réseau mobile ou NEMO (*Network Mobility*) capable de se déplacer comme une unité. Les réseaux déployés dans les véhicules (VAN : *Vehicular Area Network*) en donnent un bon exemple. Malgré le besoin de fournir un accès Internet permanent à toutes les stations localisées dans un réseau mobile, suscité soit par les fabricants de véhicules, soit par les compagnies de transport, soit par les usagers eux-mêmes, des travaux se préoccupant des problèmes spécifiques liés au déplacement des réseaux n'ont réellement vu le jour que récemment. Ces travaux ont mené à la mise en place d'un standard IETF appelé support de base de NEMO [RFC3963], permettant de supporter les réseaux mobiles dans IPv6.

Dans la première partie de ce chapitre, nous nous intéressons à la mobilité des nœuds IPv6, et la manière dont elle est gérée par le protocole Mobile IPv6.

Nous passons ensuite à décrire l'état de l'art concernant la mobilité des réseaux dans IPv6. Après une présentation succincte de la terminologie liée au sujet, nous

introduisons la notion d'enchaînement de la mobilité dans les réseaux NEMO, puis nous citons quelques exemples d'applications possibles de ces réseaux. Vient ensuite une description de la problématique posée par la mobilité des réseaux. Nous consacrons la dernière partie à la description du support de base de NEMO.

3.1 Mobilité des nœuds dans IPv6

3.1.1 Adressage IP et mobilité

L'Internet est une agglomération de réseaux partitionnés en plusieurs domaines. Un domaine représente d'ordinaire un campus, une entreprise, un fournisseur de service. Un domaine peut lui-même être divisé en sites. L'ensemble des noeuds se trouvant sur le même lien logique constitue un sous-réseau. Les noeuds sont de deux types. Ceux qui relient un sous-réseau à un autre sont des routeurs, les autres de simples stations. A chaque sous-réseau, correspond un préfixe qui permet d'identifier la position du sous-réseau dans la hiérarchie de l'Internet. Un réseau est donc un ensemble de sous-réseaux partageant le même préfixe IP et connectés à l'Internet par le biais d'un ou plusieurs routeurs externes. Tous les noeuds ayant une interface sur un sous-réseau donné ont une adresse IP correspondant au préfixe de ce sous-réseau. Cette adresse identifie à la fois la position topologique du nœud (fonction de localisation), et le noeud lui-même (fonction d'identification).

Lorsque un nœud mobile change son point d'attachement à l'Internet, il doit changer son adresse IP puisque celle-ci identifie sa position. Or, cette même adresse étant utilisé pour identifier le nœud, tout changement l'affectant aurait pour effet d'interrompre la connexion TCP en cours du nœud .

Lors des études initiales pour IPv6, il a été proposé de séparer ces deux fonctions pour pouvoir résoudre simplement les problèmes mobilité, de renumérotation de sites, multi-domiciliation. Cette proposition n'a cependant pas été retenue : en IPv6 comme en IPv4, l'adresse sert à la fois pour l'identification et la localisation. En effet, le plan d'adressage IPv6 actuellement utilisé est une extension des règles d'adressage hiérarchiques (CIDR) utilisées dans IPv4.

3.1.2 Le protocole Mobile IPv6

L'émergence de la nouvelle version du protocole Internet, IPv6, s'est accompagnée de la proposition d'un protocole standard de mobilité sur IPv6 appelé Mobile IPv6 ou MIPv6 [RFC3775]. Ce protocole, constituant l'équivalent dans IPv6 de Mobile IPv4 [RFC3344], décrit un moyen de gestion de la mobilité de terminaux IPv6. Il permet à un terminal de rester toujours joignable, quelque soit sa localisation dans l'Internet et de garder ses connexions en cours alors qu'il est entrain de se déplacer.

Commençons par définir le réseau mère d'un nœud mobile comme étant le réseau d'origine de celui-ci. Quand le nœud mobile est dans son réseau mère, il se comporte

comme un équipement stationnaire. Ce nœud mobile peut se déplacer et se connecter ainsi à l'Internet à travers un nouveau réseau. Ce réseau est appelé réseau visité.

MIPv6 décrit un mécanisme de gestion de la mobilité comportant plusieurs acteurs :

- Le nœud mobile ou MN (*Mobile Node*) : c'est un terminal IPv6 pouvant se déplacer dans l'Internet.
- L'agent mère ou HA (*Home Agent*) : quand le nœud mobile est hors de son réseau mère, d'autres nœuds peuvent être amenés à le joindre. D'où la nécessité pour ce mobile d'avoir en permanence dans ce réseau un représentant : l'agent mère du mobile.
- Le nœud correspondant ou CN (*Correspondent Node*) : c'est un terminal IPv6 avec qui le terminal mobile a ou aura une communication.

Selon Mobile IPv6, un MN est toujours joignable par son adresse principale, appelée également adresse mère appelée HoA (*Home Address*), qu'il soit attaché depuis son réseau d'origine (ou réseau mère) ou qu'il soit dans un réseau visité [RFC3775]. Tant que le MN est dans son réseau mère, les paquets à destination de l'adresse principale du MN sont routés en utilisant les mécanismes de routage IPv6 conventionnels (i.e. en fonction du préfixe de réseau) comme toute machine stationnaire.

En revanche, lorsqu'un MN se déplace, il acquière dans le réseau visité une nouvelle adresse temporaire appelée CoA (*Care Off Address*) en utilisant les mécanismes d'autoconfiguration d'IPv6. Une fois cette adresse temporaire obtenue, le MN l'enregistre auprès de son HA. Celui-ci stocke dans une table la correspondance entre l'adresse principale du MN et son adresse temporaire. Cette table est appelée la table des associations. L'enregistrement de l'adresse CoA se fait grâce à un message de mise à jour d'association appelé message BU (*Binding Update*) envoyé par le MN à son HA. Par la suite, le HA interceptera tous les paquets IPv6 destinés à l'adresse principale du MN. Ces paquets seront acheminés vers la position courante du MN. Pour ceci, le HA utilise les mécanismes d'encapsulation IPv6 [RFC2473], et adresse les paquets encapsulés à l'adresse temporaire du MN. De même, les paquets envoyés par le MN à son CN passent d'abord par le HA.

Il existe deux modes de communications dans la MIPv6. Le premier mode utilise un tunnel bidirectionnel entre le nœud MN et son HA comme décrit ci-dessus. Le deuxième mode utilise "l'optimisation du routage" qui évite le passage par le HA. Dans ce deuxième mode de communication, le MN et son correspondant s'envoient directement les paquets.

3.1.2.1 Le tunnel bidirectionnel

Tous les nœuds IPv6 ne supportent pas forcément les mécanismes de la mobilité IPv6. Alors, il peut arriver qu'un nœud n'implémentant pas MIPv6 soit un correspondant d'un nœud mobile. Dans ce cas, le correspondant n'a aucun moyen de connaître la position courante du MN (i.e. son adresse temporaire) et pour envoyer des paquets au nœud mobile, il les destine à l'adresse principale de celui-ci. Ces paquets vont être interceptés par le HA qui les encapsule pour les rediriger vers l'adresse temporaire

du MN. Lorsque le ce dernier reçoit un paquet encapsulé par son HA, il suppose que l'expéditeur original ne dispose pas de son adresse temporaire et tente de la lui faire connaître. Seulement, le CN, ne supportant pas ce mécanisme, envoie un message d'erreur ICMP. Ainsi, la communication directe ne peut avoir lieu : le CN continue à envoyer ses paquets à l'adresse principale (*cf.* figure 3.1).

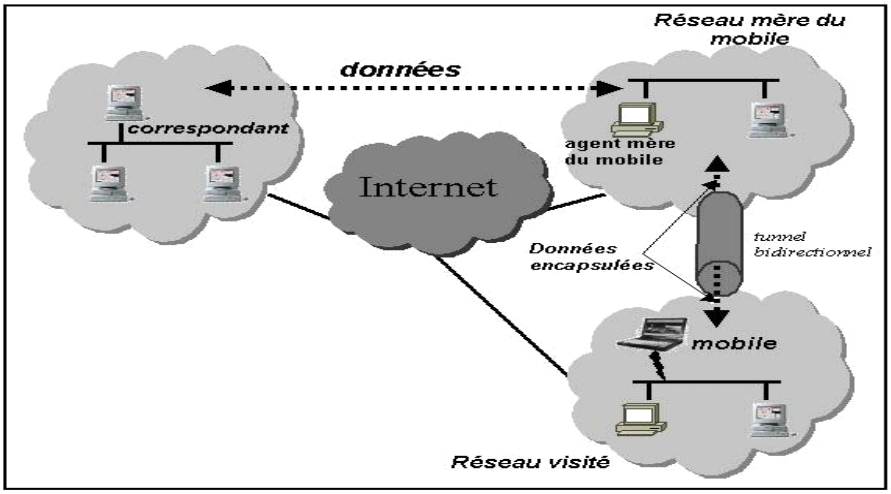


FIG. 3.1 – Le tunnel bidirectionnel de Mobile IPv6

De même, le MN ne peut pas envoyer ses paquets directement au CN car pour que ceci puisse être possible, il doit utiliser son adresse principale (qui n'est pas topologiquement correcte depuis sa position courante) comme adresse source. Ces paquets risquent d'être supprimés par les routeurs qui, en général, mettent en œuvre des politiques de sécurité interdisant la propagation de paquets ayant, comme adresse source, une adresse ne respectant pas les préfixes réseaux du réseau visité. Le MN est ainsi obligé d'utiliser son adresse temporaire qui n'est pas connue du CN. De ce fait, pour envoyer un paquet à celui-ci, le MN place son adresse principale dans le champ adresse source de l'entête IPv6 et insère celle du CN dans le champ adresse destination. Par la suite, le nœud encapsule ce paquet dans un autre paquet IPv6 où il place son adresse temporaire et l'adresse de son HA respectivement dans les champs adresses source et destination. Ainsi, le paquet est envoyé à son HA qui le désencapsule pour le rediriger vers le CN.

3.1.2.2 L'optimisation du routage

La communication utilisant le tunnel bidirectionnel est peu efficace comparée à une communication directe entre le MN et son CN. Cette notion de communication directe est connue sous le nom d'optimisation du routage. Son utilisation n'est possible que si le CN implémente les mécanismes de la mobilité IPv6. Dans ce cas, le nœud IPv6 dispose d'un cache ou table d'associations lui permettant d'enregistrer l'adresse temporaire d'un MN et de faire la correspondance avec l'adresse principale

de celui-ci. Ainsi, un MN peut enregistrer son adresse temporaire au niveau du CN et communiquer directement avec ce dernier.

3.2 Mobilité des réseaux dans IPv6

3.2.1 Définition et terminologie

La terminologie relative aux réseaux mobiles a été spécifiée dans [ERN05a]. Un réseau mobile appelé aussi réseau NEMO (*Network Mobility*) est défini comme étant un ensemble de sous-réseaux, connecté à l'Internet par l'intermédiaire d'un ou plusieurs routeurs mobiles, dont les déplacements induisent des mouvement d'ensemble du réseau NEMO tout entier. Par analogie aux stations mobiles, un routeur est dit mobile (MR : *Mobile Router*) s'il change dynamiquement son point d'ancrage (AR : *Access Router*) à l'Internet. Autrement dit, c'est un nœud mobile (MN) qui joue en plus la fonction de routeur. Comme dans le cas d'une station mobile, le point d'attachement initial est appelé le sous-réseau mère, tandis que chaque point d'attachement successif est appelé sous-réseau visité. Les interfaces d'un MR connectées sur un sous-réseau mère ou un sous-réseau étranger sont nommées interfaces externes tandis que toutes les autres interfaces sont nommées interfaces internes. La figure 3.2 montre un réseau mobile se déplaçant de son réseau mère vers un sous-réseau visité.

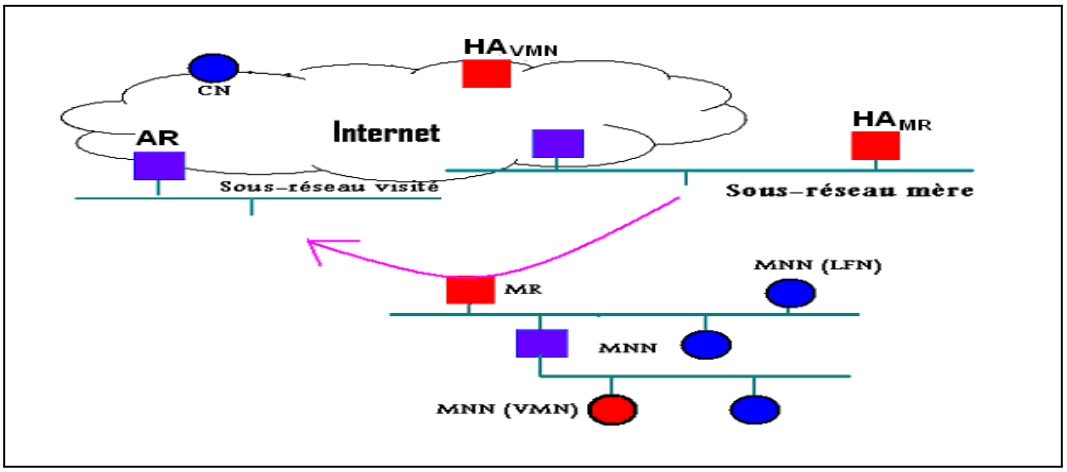


FIG. 3.2 – Terminologie des réseaux mobiles

Les réseaux mobiles ne doivent pas être confondus avec les réseaux *ad-hoc* qui sont des réseaux sans infrastructure et dont l'ensemble des noeuds sont des routeurs mobiles, avec ou sans sous-réseau attaché à leur interface interne. En effet, la configuration interne d'un réseau mobile est supposée être assez stable relativement au MR.

Le terme générique MNN (*Mobile Network Node*) désigne tout noeud localisé à l'intérieur du réseau mobile. Parmi les MNNs, il convient de différencier un nœud fixe

local résidant de manière permanente dans le réseau mobile (LFN : *Local Fixed Node*), d'un nœud mobile local appartenant au réseau mobile (LMN : *Local Mobile Node*), et d'un nœud mobile étranger n'appartenant pas au réseau mobile mais s'y attachant (VMN : *Visiting Mobile Node*). Tout noeud de l'Internet communiquant avec un ou plusieurs MNNs est considéré comme noeud correspondant (CN) du réseau mobile.

3.2.2 Emboîtement de la mobilité

L'emboîtement ou l'enchaînement de la mobilité (*Nesting*) est un nouveau type de configuration rendu possible grâce à la mobilité des réseaux. Dans le cas d'un VMN venant s'attacher à un réseau NEMO, nous faisons face à une double mobilité, celle du réseau, et celle du VMN (*cf.* figure 3.2). Dans le cas d'un routeur mobile faisant lui-même office de passerelle à un réseau mobile, qui à son tour permet l'ancrage d'un VMN (station ou routeur), nous avons trois niveaux de mobilité. La mobilité des réseaux rend théoriquement possible un nombre illimité de niveaux de mobilité. Supposons qu'un réseau mobile N2 vient s'attacher, via son routeur mobile MR2, à un autre réseau mobile N1 de routeur mobile MR1 attaché à l'Internet. La terminologie suivante est alors utilisée :

- N1 est "NEMO père" de N2 et MR1 est "MR père" de MR2
- N2 est "sous-NEMO" de N1 et MR2 est "sous-MR" de MR1
- L'ensemble {N1, N2} forme un réseau NEMO emboîté N
- MR1 est appelé "MR racine" de N

Il est à noter qu'un réseau NEMO peut voir sa topologie changer à tout moment à cause de l'emboîtement de la mobilité.

3.2.3 Exemples d'application

Les usages possibles des réseaux mobiles sont très variés. Nous pouvons citer, à titre d'exemple :

- les réseaux de capteurs déployés dans les véhicules (avions, trains, bateaux, voitures) qui ont besoin d'interagir avec des serveurs dans Internet, par exemple pour assurer la transmission de données nécessaires à la navigation.
- les réseaux d'accès déployés dans les transports publics (en particulier bus, trains et taxis) offrant une borne d'accès Internet aux passagers.
- les réseaux personnels (*Personal Area Networks*) constitués par l'ensemble des appareils électroniques de petite taille (montres, téléphones cellulaires, agendas électroniques et autres assistants personnels, appareils photo digitaux...) portés par les personnes.

Un exemple typique est celui d'une compagnie de transport ferroviaire qui offre un accès Internet permanent et ininterrompu à ses passagers par le biais de plusieurs technologies sans fil (cellulaire, IEEE 802.11b, *Bluetooth* et satellite). Le point d'ancrage dans la topologie Internet change alors fréquemment, non seulement à cause de la vitesse de déplacement du réseau, mais aussi à cause de la variété des technologies d'accès qui lui seront probablement offertes selon sa position. Cet accès permettrait

aux passagers de se connecter sur un site distant, télécharger de la musique et de la vidéo depuis n'importe quel fournisseur de service, ou de surfer sur la toile sans interruption de service en utilisant aussi bien les terminaux proposés par la compagnie que leur propres ordinateurs portables, téléphones, etc.

3.2.4 Problématique des réseaux mobiles

La problématique des réseaux mobiles a fait sommairement son apparition à l'IETF à plusieurs reprises avant de prendre véritablement son envol à partir de l'année 2000. Les concepteurs de Mobile IP [RFC3344][SOL03] proposent de gérer la mobilité des réseaux de manière similaire à celle des stations, mais ceci est présenté de manière très succincte, en partant de l'observation qu'un réseau mobile n'est rien d'autre qu'un réseau rattaché à un routeur mobile, et que seul ce dernier est tenu de changer l'adresse de son interface externe, la structure interne du réseau étant préservée lors des déplacements du MR. En effet, ces déplacements n'induisent aucun changement du point d'ancrage physique des MNNs, ceux-ci ne sont alors pas tenus de changer d'adresse. Pour ces raisons, les auteurs de MIP annoncent que le problème pourrait être résolu en appliquant l'approche MIP au MR qui n'est rien d'autre qu'un nœud mobile. A chacun de ses déplacements, il suffirait donc au MR d'obtenir une adresse temporaire MR_CoA et de l'enregistrer auprès de son HA comme dans le cas d'une station mobile.

Cette analyse simpliste n'a cependant pas été suffisamment poussée par les auteurs pour considérer les caractéristiques et les exigences spécifiques à la mobilité des réseaux. De nombreux problèmes subsistent donc. L'auteur de [ERN01] et [ERN03] a mis en évidence les insuffisances cruciales que présente Mobile IPv6, tel qu'il est défini, à supporter les stations situées derrière le MR. En effet, une fois le HA reçoit un BU provenant du MR, il ne sera capable de rediriger que les paquets destinés à ce dernier mais pas ceux destinés aux MNNs du réseau mobile.

Pour éviter ce problème, il faut donc prendre en compte le fait que la mobilité du réseau, bien qu'elle n'implique aucun changement du point d'ancrage physique des MNNs, fait de sorte que ces derniers semblent être mobiles de point de vue des CNs. On pourrait alors penser à confier à chaque MNN de gérer, de façon individuelle, l'impact des mouvements d'ensemble du réseau sur ses propres communications. Cependant, cette approche est à écarter pour plusieurs raisons. En effet, non seulement elle impose aux stations situées dans un réseau mobile d'être munies d'un support de mobilité pour pouvoir établir des communications normales, mais aussi elle nécessite d'introduire des modifications radicales sur les fonctionnalités de Mobile IP afin de l'adapter à ces stations. Ceci est évidemment du au fait qu'un MNN, à la différence d'un nœud mobile traditionnel, est incapable de détecter de façon automatique ses mouvements induits par ceux du réseau, et d'acquérir une COA, ces actions étant déclenchée par le changement du point d'attache physique du nœud en question. Il serait alors nécessaire de déployer des nouveaux mécanismes permettant de mettre les MNNs au courant des mouvements du réseau. Les messages engendrés par ces mécanismes, ajoutés aux BU envoyés périodiquement par chaque MNN à chacun de ses CNs, constituent un trafic

de contrôle important et qui risque même de congestionner le réseau mobile, ce dernier étant susceptible d'embarquer un nombre important de noeuds (de l'ordre de plusieurs centaines), chacun pouvant avoir un certain nombre de correspondants.

Pour les raisons citées ci-dessus, il s'est avéré que Mobile IP n'est pas adapté au support de la mobilité des réseaux. La communauté IETF a donc pris conscience du besoin de traiter le cas des réseaux mobiles comme un sujet à part entière. Pour éviter les interférences avec le développement de Mobile IP, elle a créé, en octobre 2002, un nouveau groupe de travail nommé NEMO (*NEtwork MObility*). Lors de sa création, ce groupe a décidé d'aborder le problème en deux étapes afin de produire une solution rapidement applicable. Dans un premier temps, il a standardisé un support de base pour la mobilité des réseaux [RFC3963]. Une solution simple y a été alors définie permettant de maintenir les sessions pour l'ensemble des MNNs, sans prendre en compte l'optimisation de routage. Dans un second temps, le groupe se doit d'étudier les problèmes d'optimisation, en particulier l'optimisation du routage (support étendu).

3.2.5 Support de base de NEMO

La solution pour le support de base a été définie en étendant le modèle MIPv6 selon des règles préalablement édictées par le groupe de travail dans un document dressant la liste des fonctions requises [ERN05b]. La règle fondamentale est de maintenir les sessions sans imposer de modifications sur les noeuds localisés derrière le routeur mobile (MNNs) et, sans optimisation de routage.

Cette solution permet la seule redirection des paquets destinés aux MNNs vers la position courante du MR. Elle consiste à établir un tunnel bidirectionnel entre le HA et le MR. Le principe de base est que tous les noeuds du réseau mobile partagent le (ou les) même préfixe d'adresse IP (MNP : *Mobile Network Prefix*).

Comme dans MIPv6, le support de base gère le problème de la mobilité en allouant deux adresses à chaque interface externe du MR (ou des MRs dans le cas où il y en aurait plusieurs). La première MR_HoA est une adresse permanente qui identifie le MR dans le sous-réseau mère. Elle identifie soit l'interface externe et a pour préfixe celui du sous-réseau mère, soit l'interface interne du MR, et elle a pour préfixe MNP comme chacun des MNNs du même réseau mobile. La seconde (MR_CoA) est temporaire, et est obtenue dans le sous-réseau visité sur lequel l'interface externe du MR prend ancrage. Le protocole établit ainsi une relation entre le préfixe MNP utilisé comme identificateur, et l'adresse temporaire MR_CoA, utilisée pour le routage. Seuls les MRs qui changent leur point d'ancrage obtiennent cette nouvelle adresse, les autres MNNs conservent leur seule adresse MNNMNP ; la gestion de la mobilité leur est ainsi transparente.

Le MR fait ensuite parvenir l'adresse temporaire primaire MR_CoA au moyen d'un message de mise-à-jour des préfixes (PBU) à son agent mère (HA). Les PBUs sont des paquets spéciaux contenant un en-tête d'extension *Mobility Header*. Lorsque HA reçoit un PBU valide (i.e. obéissant aux tests de conformité liés à la sécurité, particulièrement l'authentification de l'émetteur par son destinataire), l'entrée correspondante au MNP est ajoutée ou mise à jour dans son cache de correspondance(*Binding*

Cache). Elle instruit le HA d'encapsuler les paquets à destination des stations résidants dans le réseau mobile vers la destination effective du réseau mobile (i.e. MRc) dans la mesure où le préfixe de l'adresse de destination correspond à celui enregistré dans le cache.

Lors d'une communication entre un MNN et un CN, le CN n'a pas connaissance de l'adresse de routage temporaire MR_CoA. Les paquets sont donc envoyés normalement vers l'adresse MNNMNP du MNN et routés jusqu'au sous-réseau ayant pour préfixe MNP. Ils parviennent ainsi sur le sous-réseau mère du MR. Les paquets y sont interceptés par le HA puis encapsulés vers MR_CoA. A la réception d'un paquet encapsulé, le MR le désencapsule et le transmet sur son interface interne. Le paquet que reçoit le MNN ne contient donc plus MR_CoA ; l'opération lui est ainsi transparente. Dans le sens inverse, les paquets sont également encapsulés du MR à son HA.

3.3 Conclusion

Dans ce chapitre, nous nous sommes intéressés à la mobilité des réseaux dans IPv6, qui est une forme de mobilité plus récemment invoquées par les travaux de recherches que la mobilité classiques des hôtes.

Après avoir décrit le protocole Mobile IPv6, le standard de l'IETF pour la gestion de la mobilité des nœuds IPv6 mobiles, nous avons expliqué les enjeux supplémentaires introduits par la mobilité des réseaux par rapport à celle des nœuds. Nous avons ensuite enchaîné avec une présentation brève du support de base de NEMO, le standard IETF pour la gestion des réseaux IPv6 mobiles. Ce support est une extension du protocole Mobile IPv6, il utilise un tunnel entre le routeur mobile et son agent mère pour acheminer le trafic entre les nœuds du réseau mobile et le reste de l'Internet. Ainsi, il permet à ces nœuds de rester joignables à des adresses IP fixes et de maintenir leurs communications en cours sans se rendre compte de la mobilité du réseau.

Il est cependant à noter que ce support ne permet pas de gérer les communications *multicast* des nœuds d'un réseau mobile. En effet, il utilise le préfixe de réseau contenu exclusivement dans les adresses *unicast* pour la redirection des paquets. Pour gérer les sessions *multicast* dans les environnements NEMO, de nouveaux mécanismes sont alors nécessaires.

Chapitre 4

Multicast et Mobilité des Réseaux dans IPv6

A l'ère des services multimédia, tels que la télé et la radio sur Internet, les vidéo-conférences, les jeux en ligne et les applications collaboratives, le *multicast* IP est extrêmement prometteur, vu son apport reconnu pour les communications de groupe. De plus, la migration vers la nouvelle génération du protocole IP [RFC2460] favorise le déploiement d'un service *multicast* natif dans l'Internet.

Par ailleurs, avec le développement et l'amplification des nouveaux services de mobilité, il est évident que les utilisateurs mobiles d'Internet vont s'attendre à avoir accès aux services et applications disponibles via les réseaux filaires, en particulier les applications multimédia et de collaboration.

Or, bien qu'ils soient conçus de façon à permettre la gestion dynamique des abonnements aux groupes, les protocoles *multicast* actuels sont dépourvus de support de mobilité. En effet, contrairement à ce que l'on pourrait penser, abonnement dynamique et mobilité sont deux problèmes qui impliquent des besoins différents, ne pouvant pas être satisfaits par les mêmes mécanismes. Plusieurs efforts de recherches ont été alors fournis, Dans le but de permettre un couplage entre *multicast* et mobilité dans l'Internet, en particulier pour la nouvelle génération du protocole IP (IPv6). Le protocole MIPv6 [RFC3775] fournit deux solutions de base pour la gestion des nœuds *multicast* mobiles dans IPv6 : le *tunnelling* bidirectionnel et l'enregistrement à distance. D'autres alternatives aux solutions de l'IETF ont été aussi proposées.

Avec l'apparition d'une nouvelle forme de mobilité IP qu'est la mobilité des réseaux ou NEMO (*cf.* chapitre 3), le problème d'association entre *multicast* et mobilité a pris de nouvelles dimensions. En effet, fournir un support *multicast* est aussi important dans le cas de réseaux mobiles que dans le cas de nœuds mobiles, puisque les utilisateurs potentiels de ce type de réseau auront aussi besoin d'accéder aux applications multimédia et de collaboration basées sur les communications de groupe.

Le support de base de NEMO [RFC3963] permet aux nœuds dans un réseau mobile de rester joignables à des adresses IP fixes et de maintenir leurs communications *unicast* en cours sans se rendre compte de la mobilité du réseau (*cf.* chapitre 3, section 3.2.5). Cependant, ce support ne permet pas de gérer les communications *mul-*

ticast. Dans le but de combler ce manque, une solution unique a été déjà proposée [JAN04]. Cette solution se base sur le déploiement, dans les réseaux mobiles, du *proxying* MLD qui est une technique d'acheminement *multicast* originellement conçue pour s'en passer du déploiement d'un protocole de routage *multicast* dans les réseaux de bordure à topologie simplifiée [FEN04].

En premier lieu, nous présentons une vue d'ensemble sur l'état de l'art concernant l'association entre *multicast* et mobilité des nœuds dans IPv6. Nous nous intéressons en particulier aux solutions de l'IETF proposées à ce propos et nous les comparons. Nous citons aussi quelques solutions alternatives.

Nous décrivons ensuite l'unique solution proposée jusqu'à maintenant pour supporter le *multicast* dans les réseaux NEMO, basée sur l'utilisation du *proxying* MLD. Nous terminons ce chapitre par une critique argumentée qui montre les problèmes que présente cette solution, allant de la redondance de trafic jusqu'au risque de création de boucles de transmission.

4.1 *Multicast* pour les nœuds IPv6 mobiles

Le scénario de *handover* est particulièrement défiant dans le cas des communications *multicast*. En effet, le déplacement d'un membre *multicast* induit l'invalidation de la branche *multicast* qui le dessert, puisque celle-ci est construite relativement à la position de ce membre. De même, le déplacement d'une source *multicast* rend invalide tout l'arbre de livraison si cet arbre est basé à la source. Un autre problème est celui de permettre à un hôte mobile de rejoindre de nouvelles sessions *multicast* en tant que membre ou source alors qu'il est loin de son réseau mère.

Les protocoles de routage *multicast* proposés par la communauté d'Internet supposent que les membres et les sources *multicast* sont stationnaires. Afin de satisfaire les exigences supplémentaires induites par la mobilité des hôtes *multicast*, plusieurs propositions ont été fournies.

4.1.1 Gestion des communications *multicast* par Mobile IPv6

Le groupe de travail MIP6 de l'IETF a proposé deux solutions de base faisant partie de la spécification de Mobile IPv6 [RFC3775], et permettant à un nœud IPv6 mobile de recevoir et d'envoyer du trafic *multicast*. Ces solutions, connues dans la littérature sous les noms d'enregistrement à distance (RS : *Remote Subscription*) et de *tunnelling* bidirectionnel (BT : *Bi-directional Tunnelling*), ont été en réalité invoquées pour la première fois par Mobile IPv4 [RFC3344]. Le protocole Mobile IPv6 reprend alors les mêmes mécanismes pour assurer la gestion des sessions *multicast* des nœuds IPv6 mobiles.

Selon le protocole MIPv6, lorsqu'un noeud *multicast* mobile est dans son réseau d'origine (réseau mère), il se comporte comme toute station IPv6 standard : il utilise l'infrastructure de son réseau mère pour rejoindre les sessions *multicast*, recevoir et/ou envoyer du trafic *multicast*. Si, par contre, le nœud se déplace vers un réseau étranger,

il utilise l'une des deux approches citées ci-dessus (RS ou BT) pour maintenir ses communications *multicast* en cours et, éventuellement, participer à de nouvelles sessions *multicast* à partir du réseau visité.

4.1.1.1 Le *tunnelling* bidirectionnel

Selon cette approche, un nœud mobile attaché à un réseau étranger utilise l'infrastructure *multicast* de son réseau mère pour envoyer et recevoir du trafic *multicast*, et ce grâce au tunnel bidirectionnel de MIPv6 établi entre le MN et son HA. Notons que ce tunnel est lui-même le tunnel classique de MIPv6 utilisé pour la redirection le trafic *unicast* (*cf.* section 3.1.2 du chapitre 3). En effet, l'adresse *unicast* du HA et l'adresse temporaire (CoA) du nœud *multicast* mobile définissent les extrémités de ce tunnel.

4.1.1.1.1 Récepteur *multicast* mobile : le tunnel bidirectionnel permet au MN d'utiliser l'infrastructure *multicast* de son réseau mère pour recevoir du trafic *multicast* alors qu'il est dans un réseau étranger. Pour ce faire, le HA maintient une vue à jour des sessions *multicast* dont le MN est membre, et lui fait suivre les paquets *multicast* associés via le tunnel. La mise à jour des informations d'abonnements se fait grâce au protocole MLD déroulé de part et d'autre du tunnel. En particulier, le HA envoie, via le tunnel, des messages de recensement MLD périodiques au MN, et ce dernier utilise le tunnel dans le sens inverse pour répondre par des rapports d'abonnements MLD. Le MN envoie aussi des rapports d'abonnement MLD non sollicités au HA via le tunnel pour rejoindre (ou quitter) des nouvelles sessions *multicast*. Le HA doit implémenter soit les fonctionnalités d'un routeur *multicast*, soit celles d'un *proxy* MLD. Dans le deuxième cas, il s'abonne (ou désabonne) aux sessions *multicast* auprès du routeur *multicast* local sur le lien mère au nom du MN. Ainsi, toutes les branches *multicast* sont construites vers le réseau mère, et les paquets *multicast* suivent un chemin triangulaire passant par le HA pour être transmis au MN.

Lorsque le MN se déplace vers un nouveau réseau étranger, il n'a pas besoin de re-adhérer aux sessions *multicast* dont il est membre, puisque le HA est déjà informé de l'abonnement. Dès que le HA reçoit le BU contenant la nouvelle adresse CoA du MN, il est capable de lui acheminer les paquets *multicast* qui le concernent.

4.1.1.1.2 Source *multicast* mobile : avec l'approche de *tunnelling* bidirectionnel de Mobile IPv6, une source *multicast* mobile entrain de visiter un réseau étranger utilise l'infrastructure de son réseau mère pour envoyer du trafic *multicast*. Pour ce faire, elle encapsule tout paquet *multicast* sortant et l'envoie à son HA via le tunnel bidirectionnel MN-HA. Notamment, elle met son adresse HoA dans le champ adresse source du paquet *multicast*, et son adresse CoA dans l'en-tête *unicast* externe d'encapsulation. Le HA désencapsule alors ce paquet et le transmet sur l'arbre de livraison *multicast* associé. Ainsi, la mobilité de la source est gardée transparente vis-à-vis des membres et des routeurs intermédiaires, puisque les paquets *multicast* sont toujours transmis avec la même adresse source (HoA), sur le même arbre de livraison.

4.1.1.2 L'enregistrement à distance

Selon l'approche d'enregistrement à distance, le MN utilise l'infrastructure *multicast* du réseau étranger visité pour recevoir et envoyer du trafic *multicast*.

4.1.1.2.1 Récepteur *multicast* mobile : après chaque *handover*, le MN doit re-adhérer à toutes les sessions *multicast* dont il est déjà membre depuis sa nouvelle position. Pour ce faire, il utilise son adresse temporaire CoA pour envoyer des rapports d'abonnement MLD au routeur *multicast* local sur le lien visité. Des branches de livraison *multicast* associées à ces sessions sont alors construites vers la nouvelle position du MN, permettant à ce dernier de continuer à recevoir le trafic *multicast* correspondant. De plus, le MN se comporte désormais comme tout autre nœud fixe sur le lien visité : il rejoint (ou quitte) les sessions *multicast* via le routeur *multicast* local situé sur ce lien. En particulier, il répond aux messages de recensement MLD envoyés par ce routeur. Il peut aussi envoyer localement des messages MLD non sollicités pour rejoindre (ou quitter) de nouvelles sessions *multicast*.

Notamment, le MN utilise son adresse temporaire CoA dans tous les messages MLD qu'il envoie sur le lien visité.

4.1.1.2.2 Source *multicast* mobile : selon l'approche d'enregistrement à distance, le MN utilise l'infrastructure du réseau étranger qu'il visite pour envoyer des paquets *multicast*. Le routeur *multicast* local sur le lien visité est alors responsable de délivrer les paquets sur l'arbre *multicast*. Notons que la CoA doit être utilisée comme adresse source de chaque paquet *multicast* sortant pour éviter les problèmes éventuels de filtrage (*ingress filtering*). Ainsi, contrairement à l'approche du *tunnelling* bidirectionnel, l'arbre de livraison *multicast* sera construit à partir d'états utilisant la CoA et non la HoA.

4.1.1.3 Comparaison des *tunnelling* bidirectionnel et de l'enregistrement à distance

L'approche de *tunnelling* bidirectionnel a l'avantage de ne pas mettre à jour l'arbre de livraison à chaque mouvement du MN. De plus, elle permet de garder la mobilité d'une source *multicast* transparente vis-à-vis des récepteurs. Cependant, elle introduit un routage triangulaire non optimal des paquets *multicast*, et des opérations d'encapsulation/désencapsulation au niveau du MN et du HA. Ce dernier risque même d'être surchargé si le nombre de MNs qu'il dessert devient important. De plus, la latence d'adhésion aux groupes *multicast* est augmentée par l'étape préalable d'enregistrement de l'adresse CoA du MN auprès de son HA et par l'acheminement des rapports MLD vers ce dernier. Par ailleurs, un réseau étranger visité par plusieurs MNs associés au même HA et abonnés à un même groupe voit arriver un nombre égal de copies de chaque paquet *multicast* destiné au groupe. Ce problème est connu sous le nom de convergence de tunnel. Quand le nombre de MNs devient très important, ceci peut

même conduire à un problème de congestion. Cette approche souffre aussi de son fondement sur un point de vulnérabilité critique qui est le HA.

Comparée à l'approche de *tunnelling* bidirectionnel, l'enregistrement à distance offre un routage optimal des paquets *multicast* [JEL03a]. De plus, la latence d'adhésion aux groupes est réduite grâce à l'absence de la phase d'enregistrement de la CoA auprès du HA et à l'utilisation d'un routeur *multicast* local pour s'abonner aux groupes. Cependant, l'enregistrement à distance se prête mal au cas des nœuds qui se déplacent à une vitesse élevée. En effet, outre la signalisation importante engendrée par les mises à jour fréquentes de l'arbre de livraison *multicast*, une consommation inutile de la bande passante a lieu, puisque le MN, après avoir rejoint l'arbre *multicast* depuis un réseau visité, quitte rapidement ce réseau, mais le trafic *multicast* continue à être acheminé vers le lien visité jusqu'à expiration de l'information d'abonnement au groupe associée. Par ailleurs, cette approche suppose que chaque réseau étranger soit muni de routeur *multicast*. En d'autres termes, elle ne se base pas sur le HA comme nœud critique mais, en contrepartie, elle est vulnérable à l'absence de service *multicast* dans les réseaux visités. En considérant l'état de déploiement du *multicast* dans IPv4, ceci parait très contraignant. Cependant, avec l'arrivée de la nouvelle génération de l'Internet, il est attendu que le service *multicast* soit déployé à grande échelle.

L'approche d'enregistrement à distance est adaptée au cas d'une source mobile utilisant un arbre partagé. En effet, la source mobile envoie tous simplement le trafic *multicast* à la racine de l'arbre (exemple : le point de rendez vous dans PIM-SM), qui les transmet aux récepteurs. La mobilité de la source est partiellement transparente aux récepteurs. En effet, en utilisant sa CoA pour l'envoi des paquets *multicast*, la source ayant effectué un *handover* va apparaître aux récepteurs comme étant une nouvelle source. Notons que les protocoles de couches supérieures peuvent cacher ceci à l'utilisateur, mais au niveau réseau la source est vue comme une nouvelle source. Si, par contre, un arbre basé à la source est utilisé, la réception du trafic *multicast* est interrompue après un *handover* puisque l'arbre reste enraciné à l'ancienne position de la source. Ainsi, cette approche pose des problèmes insolubles aux protocoles de routage *multicast* chargés de construire les arbres de livraison dans le réseau. En effet, son problème majeur est qu'elle suppose que les routeurs et les récepteurs sont capables d'interpréter le trafic provenant d'une nouvelle CoA comme étant issu de la même source *multicast*. De plus, les routeurs *multicast* intermédiaires doivent effectuer des mises à jour pour optimiser les chemins de livraisons *multicast* et pour éviter le rejet des paquets *multicast* contenant la CoA comme adresse source suite au test RPF. Ainsi, l'arbre *multicast* basé à la source doit être reconstruit en entier. Durant la construction de l'arbre, la source mobile ne pourra pas commencer à envoyer les données *multicast*, ce qui mène à une interruption du service *multicast*. De plus, dans le modèle SSM, les membres du groupe adhèrent souvent à un groupe et à une source spécifique, ce qui suppose qu'à chaque changement de localisation du mobile, tout le processus d'adhésion recommence avec une nouvelle adresse de source et que l'arbre de livraison spécifique à la source soit reconstruit [JEL03b].

Rappelons que le *tunnelling* bidirectionnel résout simplement les problèmes reliés

aux sources *multicast* mobiles. En effet, indépendamment de la nature de l'arbre utilisé (partagé ou basé à la source), la source mobile envoie ses données encapsulées vers son agent mère. L'adresse source du datagramme *multicast* (c'est-à-dire l'en-tête interne du paquet encapsulé) doit être égale à l'adresse mère (HoA) de la source mobile. Ainsi, la mobilité de la source reste transparente aux récepteurs, qui continuent à recevoir des paquets *multicast* provenant de la même adresse quelque soit la position de la source.

4.1.2 Les solutions alternatives

Quelques solutions alternatives aux solutions de l'IETF ont été proposées ayant pour but d'optimiser la livraison des données *multicast* depuis et vers les nœuds mobiles. Cependant, la plupart de ces propositions ont été proposées pour IPv4, mais peuvent parfois être étendues à IPv6. A titre d'exemple, les protocoles MoM (*Mobile Multicast Protocol*) [TIM97] et RBMoM (*Range-Based Mobile Multicast*) [LIN02] ont été proposé pour résoudre le problème de convergence de tunnel. Ces protocoles ont été proposés pour IPv4 et ne peuvent pas être directement étendu à IPv6, puisqu'ils utilisent l'entité agent relais (*Foreign Agen*t) définie par Mobile IPv4 et qui n'existe pas dans Mobile IPv6.

Un autre protocole appelé MMA (*Multicast by Multicast Agent*) a été proposé dans [Hee00]. Ce protocole introduit deux nouvelles entités appelées agent *multicast* (MA : *Multicast* Agent) et agent transmetteur (MF : *Multicast Forwarder*). Un MF est chargé de transmettre les paquets *multicast* au MA situé dans le réseau étranger (qui les délivre en *multicast* natif sur son lien local). Initialement, le MF d'un réseau est le MA lui meme. Lorsque le hôte mobile atteint un réseau dont le MA est non desservi par un MF, le MF qui desservait le dernier réseau visité par le nœud juste avant son déplacement devient le MF du MA actuel. Si, au contraire, le nouveau MA est déjà desservi par un MF, il choisit entre son MF actuel et celui associé au réseau d'où vient le MN. Cette proposition peut aussi bien être appliquée avec IPv4 qu'avec IPv6.

L'architecture LAR (*Logical Addressing and Routing*) proposée dans [NOE02] a pour objectif de fournir une communication *multicast* efficace aux nœuds IPv6 mobiles. A côté de l'adresse de routage IP traditionnelle, le protocole LAR introduit un nouveau type d'adresse appelée adresse LAR. C'est une adresse logique utilisée pour identifier des entités réseau logiques telles que les routeurs LAR et les nœuds LAR. Comparée à l'adresse IP classique, l'adresse LAR reste inchangée malgré la mobilité du host LAR.

Ces deux dernières solutions définissent des mécanismes innovants afin d'optimiser la livraison *multicast* aux nœuds mobiles, mais elles souffrent tous de vrais problèmes de déploiement. En effet, elles ne passent pas à l'échelle avec la taille de l'Internet. De plus, les protocoles qu'elles définissent exigent un effort de collaboration entre des routeurs pouvant appartenir à différentes entités administratives. La nature de cette collaboration est telle que ces entités peuvent ne pas être prêts à coopérer.

Par ailleurs, dans le but de fournir un support aux sources IPv6 mobiles dans le cadre des sessions SSM, les auteurs de [JEL02a] et [JEL02b] ont proposé d'ajouter une sous-option à l'option IPv6 "*Binding Destination Option*". La nouvelle option, appelée "*SSM Source handover Notification*", est utilisée pour notifier les récepteurs *multicast* de la nécessité de se réinscrire au nouveau canal SSM après chaque *handover* de la source.

Lorsque la source se déplace à un nouveau réseau étranger, elle reçoit une nouvelle CoA (nCoA), qui est différente de son ancienne CoA (oCoA). Pour éviter l'interruption du service *multicast*, la source mobile envoie à tous les récepteurs concernés un message BU contenant la sous-option *Source handover Notification* pour les informer de sa nCoA. A la réception de cette notification explicite acheminée via l'ancien arbre de livraison *multicast* enraciné au réseau étranger précédent, les récepteurs *multicast* déclenchent la reconstruction du nouveau arbre basé à la source en envoyant des nouveaux rapports d'abonnements MLD contenant la nCoA au lieu de la oCoA. Ainsi, un nouvel arbre référencé par (nCoA, G) est construit à la place de l'ancien arbre (oCoA, G), et la source pourra utiliser sa nCoA pour envoyer du *multicast* natif via cet arbre. Comparée au *tunnelling* bidirectionnel, cette approche permet de garder le routage optimal en évitant le passage par le HA. Cependant, un *handover* de la source induit la reconstruction de tout l'arbre *multicast*, et il n'est pas transparent aux récepteurs.

4.2 *Multicast* pour les réseaux IPv6 mobiles

Fournir un service *multicast* dans un réseau mobile revient à permettre à chacun de ses nœuds de se comporter comme membre de groupe *multicast* ou comme source envoyant du trafic *multicast* de façon normale et indépendante des mouvements éventuels du réseau.

En premier lieu, il faut assurer le maintien des sessions *multicast* en cours et l'acheminement du trafic *multicast* entrant et sortant suite à un changement de la position du réseau.

En deuxième lieu, les MNNs doivent pouvoir rejoindre de nouvelles sessions *multicast* quand le réseau est loin de sa position d'origine.

Enfin, tous les mécanismes de gestion de la mobilité du réseau doivent s'effectuer de manière transparente aux MNNs, ces derniers pouvant même être dépourvus de tout support de mobilité. C'est précisément ce dernier point qui rend le problème de la gestion des nœuds *multicast* dans un réseau mobile un problème plus compliqué que celui de la gestion des nœuds *multicast* mobiles. Il est pourtant à noter que le premier problème hérite du deuxième, et que, par conséquent, les solutions pour hôtes *multicast* mobiles peuvent être exploitées en combinaison avec d'autres mécanismes pour gérer les hôtes *multicast* à l'intérieur des réseaux NEMO.

Une solution unique a été proposée jusqu'à présent pour offrir un service *multicast* pour les réseaux NEMO. Elle est basée sur l'utilisation du *proxying* MLD.

4.2.1 Solution basée sur le *proxying* MLD

Les auteurs de [JAN04] proposent d'utiliser l'approche du *proxying* MLD [FEN04] pour fournir un support aux réseaux NEMO. Le *proxying* MLD est une technique d'acheminement *multicast* récemment proposée par l'IETF dans le but d'éviter le déploiement de routeurs *multicast* dans un réseau de bordure à topologie simplifiée.

Contrairement aux protocoles de routage *multicast* basés sur une construction dynamique d'un arbre de livraison par session *multicast*, la technique du *proxying* MLD utilise une configuration manuelle et préalable d'un arbre commun formé par les *proxies* MLD et enraciné en un routeur de bordure du réseau, uniquement ce dernier étant censé implémenter un protocole de routage *multicast*. Les échanges locaux de messages MLD entre les différents *proxies* leur permettent de collecter des informations d'abonnements relatives aux sessions *multicast* afin d'éviter l'inondation inutile de l'arbre préétabli par les différents flux *multicast*. La figure 4.1 illustre le schéma général du déploiement du *proxying* MLD dans un réseau de bordure.

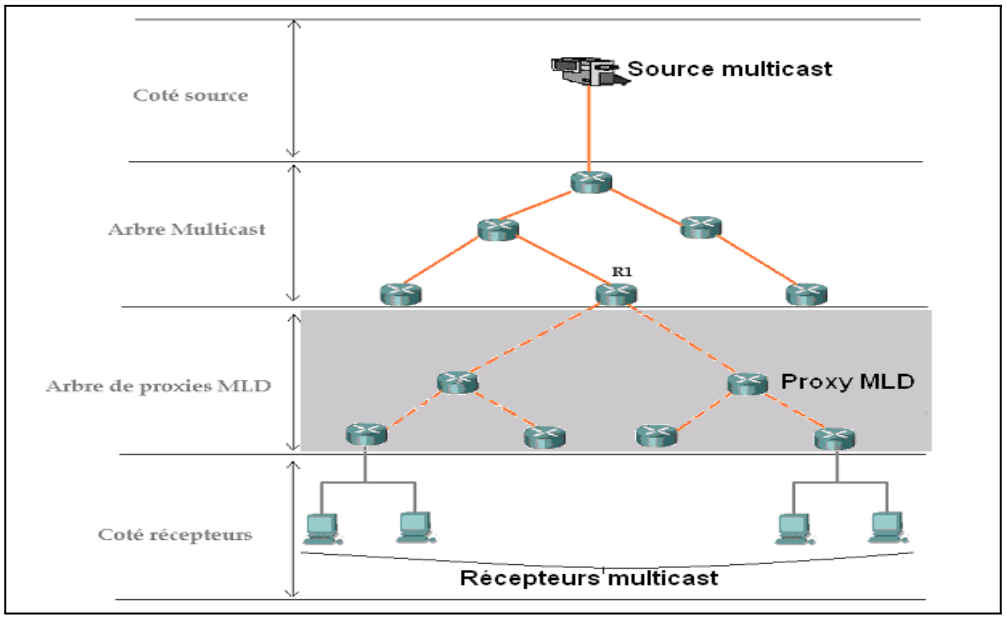


FIG. 4.1 – Déploiement du MLD *proxying* dans un réseau de bordure

Dans ce schéma, le routeur de bordure R1 est responsable de rejoindre les arbres *multicast* aux noms des récepteurs à l'intérieur du réseau de bordure contenant les récepteurs *multicast*.

4.2.1.1 Configuration d'un *proxy* MLD

Un *proxy* MLD est un nœud qui possède une interface *upstream* unique et une ou plusieurs interfaces *downstream*. Alors qu'il agit en tant que routeur MLD standard sur chaque interface *downstream* (appelée aussi interface routeur), il implémente la

partie hôte du protocole MLD sur son interface *upstream* (interface hôte). Le lien auquel est attaché cette dernière interface doit alors être muni d'un routeur MLD. Notons qu'une interface donnée peut être ni *upstream* ni *downstream* (c'est-à-dire démunie de tout support MLD), mais qu'elle ne peut pas être les deux à la fois. La figure 4.2 montre un routeur configuré en tant que *proxy* MLD. Les liens auxquels sont connectés les interfaces *downstream* du *proxy* hébergent des hôtes *multicast* standard, tandis que l'interface *upstream* connecte le *proxy* à un routeur MLD (routeur *upstream* du proxy).

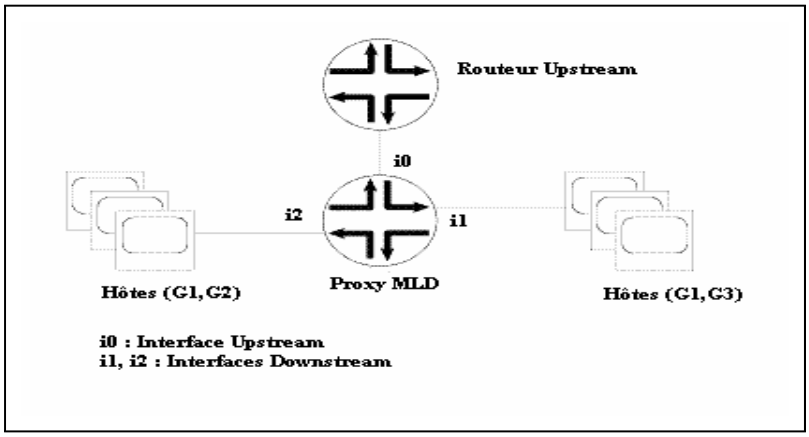


FIG. 4.2 – Cofiguration d'un *proxy* MLD

Chacun des hôtes *multicast* interagit avec le *proxy* MLD, jouant le rôle du routeur MLD sur son lien local, par l'intermédiaire de messages MLD (*cf.* chapitre 1, section 1.3.1). De même, étant configuré en tant que hôte MLD sur son interface *upstream*, le *proxy* est capable d'échanger des messages MLD via cette interface avec son routeur MLD *upstream*. Cependant, à la différence des hôtes *multicast* standard se servant du protocole MLD pour rejoindre et quitter les sessions *multicast* à leur propre compte, le *proxy* MLD, comme son nom l'indique, joue le rôle de mandataire qui fait "remonter" les informations de gestion des groupes relatives aux hôtes *downstream* vers son routeur MLD *upstream*. Par exemple, dans le cas de la figure 4.2, le *proxy* MLD reçoit des rapports d'abonnement MLD à G1 et G3 sur l'interface i1, et à G1 et G2 sur l'interface i2. Il envoie alors des rapports d'abonnement à G1, G2 et G3 via son interface *upstream* i0.

Pour assurer ce rôle de mandataire, le *proxy* doit obéir à des lois spécifiques d'envoi des rapports MLD via son interface *upstream*. Nous détaillerons ces lois plus loin dans ce chapitre (*cf.* section 4.2.1.4).

De plus, le *proxy* MLD maintient une base de données contenant les informations d'abonnements aux groupes collectées sur ses interfaces *downstream*. Il utilise ces informations pour l'acheminement ultérieur des paquets *multicast*.

4.2.1.2 Configuration de l'arbre des *proxies* MLD

Un hôte desservi par une interface *downstream* d'un *proxy* MLD peut aussi bien être une station *multicast* standard qu'une interface *upstream* d'un autre *proxy* MLD. Il en résulte la possibilité d'organiser un ensemble de *proxies* MLD en un arbre où chaque nœud est relié via son interface *upstream* à l'une des interfaces *downstream* de son père. Une telle configuration permet de fournir un support de livraison *multicast* basé sur le *proxying* MLD, dans un réseau de bordure à topologie simplifiée [FEN04]. Les auteurs de [JAN04] proposent d'utiliser l'approche du *proxying* MLD pour fournir un support *multicast* dans les réseaux NEMO.

4.2.1.2.1 Configuration manuelle de l'arbre : pour déployer le *proxying* MLD dans un réseau donné, il suffit de configurer manuellement un sous-ensemble de ses routeurs en des *proxies* MLD de manière à former un arbre desservant tous les liens du réseau (arbre recouvrant). Prenons comme exemple le réseau NEMO représenté par la figure 4.3 et expliquons comment l'approche du *proxying* MLD peut lui être appliquée.

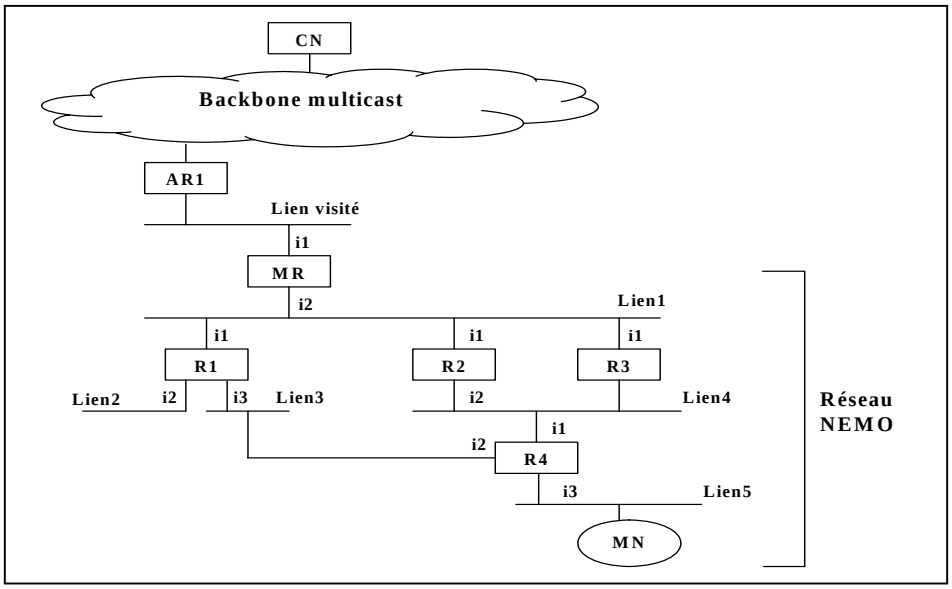


FIG. 4.3 – Réseau mobile comprenant des boucles

Ce réseau NEMO comprend 5 LAN IPv6 différents (du lien1 au lien5), interconnectés entre eux par des routeurs fixes (de R1 à R4). Le routeur mobile MR relie l'ensemble au lien étranger A desservi par le routeur d'accès AR1. AR1 est un routeur *multicast*.

Les routeurs du réseau mobile doivent être configurés manuellement de façon à former un arbre de *proxies* MLD de racine MR. Cette configuration comporte deux tâches :

1. Sélectionner les routeurs à configurer en MLD-*proxies*.
2. Désigner, pour chaque *proxy* MLD, une interface *upstream* unique et une ou plusieurs interfaces *downstream*. Particulièrement, l'interface externe du MR doit être configurée en interface *upstream*.

Le tableau suivant représente une configuration possible du réseau NEMO de la figure 4.3 :

	MLD-*proxy* ?	It. *upstream* (hôte MLD)	It. *downstream* (routeur MLD)
MR	OUI	i1	i2
R1	OUI	i1	i2,i3
R2	OUI	i1	i2
R3	NON	Aucune	Aucune
R4	OUI	i1	i3

Il est à noter que, selon cette configuration, R3 ne doit dérouler ni la partie hôte ni la partie routeur du protocole MLD sur ses interfaces. D'autre part, R4 est configuré en MLD *proxy* mais ne déroule pas le protocole MLD sur son interface i2. L'arbre résultant de cette configuration est représenté par la figure 4.4.

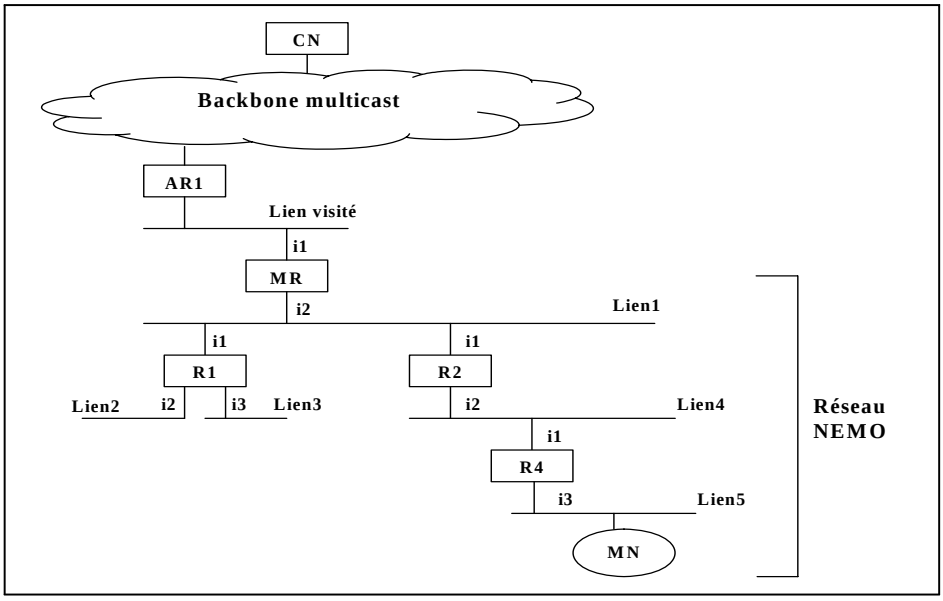


FIG. 4.4 – Configuration d'un réseau mobile en arbre de *proxies* MLD

Ainsi, de point de vu routage *multicast*, la topologie est celle d'un arbre où toutes les boucles ont été éliminées. Tout trafic *multicast* destiné à des MNNs leur sera livré en empruntant des chemins inclus dans cet arbre. Notons cependant que ceci n'a aucun impact sur le routage *unicast* au sein du réseau mobile. Par exemple, R3 continue à acheminer les paquets *unicast* de façon normale.

4.2.1.2.2 Tolérance aux pannes : la façon la plus simple de déployer le *proxying* MLD est d'établir une structure d'arbre figée par la sélection des *proxies* MLD et de leurs interfaces *upstream* et *downstream* comme nous avons expliqué dans la section précédente. Cependant, cette méthode de configuration totalement manuelle ne fournit aucune tolérance aux pannes des liens ou des routeurs puisque les *proxies* ne peuvent pas reconstruire un arbre *multicast* en fonction de l'état du réseau. Ainsi, la panne d'un *proxy* d'une hiérarchie donnée entraîne l'isolation de tous les *proxies* du niveau inférieur, et, par conséquent, toute la branche d'arbre située sous ce *proxy* ne peut plus être desservie en trafic *multicast*. Dans notre exemple, si R2 tombe en panne, les paquets *multicast* ne peuvent plus atteindre les liens L1 et L5 puisque R3 n'appartient pas à l'arbre des *proxies* MLD et ne peut donc acheminer du trafic *multicast*.

Il est cependant possible de renforcer la tolérance aux pannes en autorisant la coexistence de différents *proxies* MLD connectés par des interfaces *downstream* (routeur) à un même lien tout en permettant à un seul d'entre eux (le *proxy* acheminer), à un instant donné, de propager les informations d'abonnements aux groupes relatives au lien via son interface *upstream* et de livrer du trafic *multicast* sur ce lien. Cette dernière mesure vise, notamment, à garder une structure d'arbre et à éviter la redondance du trafic sur les liens. Pour le choix du *proxy* acheminer, les auteurs de [FEN04] proposent d'exploiter le procédé d'élection de routeur recenseur (*Querier*) déjà défini par le protocole MLD : le *proxy* élu comme recenseur MLD sur le lien sera alors lui-même le *proxy* acheminer. Si le recenseur/acheminer d'un lien tombe en panne, un autre parmi les *proxies* MLD sur ce lien est automatiquement élu à sa place. Ce dernier commence immédiatement à dérouler la partie hôte du protocole MLD sur son interface *upstream* afin de propager les informations d'abonnements aux groupes relatives au lien en question (qu'il collectait même avant d'être élu acheminer) vers son routeur *upstream*. De plus, il est désormais responsable de propager les paquets *multicast* sur le lien pour lequel il a été élu recenseur/acheminer en se basant sur sa base de données des abonnements.

Revenons à l'exemple de la figure 4.3. Une deuxième configuration possible de ce réseau en *proxying* MLD est représentée par le tableau ci-dessous. Les changements par rapport à la configuration précédente sont indiqués par des parenthèses.

	MLD-*proxy* ?	It. *upstream* (hôte MLD)	It. *downstream* (routeur MLD)
MR	OUI	i1	i2
R1	OUI	i1	i2,i3
R2	OUI	i1	i2
R3	(OUI)	(i1)	(i2)
R4	OUI	i1	(i2),i3

Dans cette configuration, chaque routeur du réseau a été configuré en *proxy* MLD dont toutes les interfaces autres que l'interface *upstream* sont configurées en interfaces *downstream*. Une telle configuration maximise la tolérance aux fautes du réseau puisque tous les routeurs sont potentiellement aptes d'acheminer du trafic *multicast*.

Supposons par exemple que l'adresse IPv6 assignée à l'interface i3 de R1 soit plus

petite que celle assignée à l'interface i2 de R4, et que l'adresse IPv6 de l'interface i2 de R2 soit plus petite que celle de l'interface i2 de R3. Selon la spécification du protocole MLD [RFC2710][RFC3810], R1 et R2 seront alors élus recenseur/acheminеur respectivement pour les liens L3 et L4. Il en résulte un arbre équivalent à celui représenté par la figure 4.4.

Si maintenant R1 et R2 tombent en panne, R4 et R3 seront automatiquement élus recenseur/acheminеur respectivement pour L3 et L4. Le nouvel arbre *multicast* résultant est représenté par la figure 4.5. Les sessions *multicast* en cours sont maintenues.

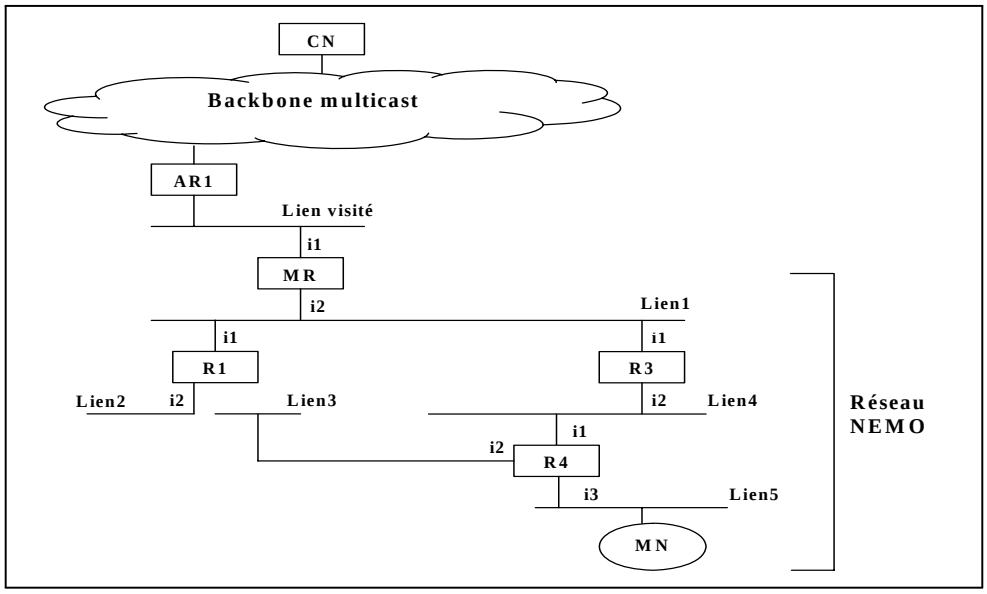


FIG. 4.5 – Reconfiguration de l'arbre des *proxies* MLD après panne

Il est à noter que lorsque plusieurs équipements sur le même lien sont configurés en *proxies* MLD, il est important de s'assurer que la topologie *multicast* globale résultante à l'intérieur du réseau mobile forme toujours un arbre enraciné au MR, indépendamment du résultat de l'élection du Forwarder déroulée sur chaque lien.

4.2.1.3 Gestion et échange des informations d'abonnements aux groupes

Rappelons qu'un *proxy* MLD joue le rôle d'un routeur MLD sur chacune de ses interfaces *downstream*, et d'un hôte MLD sur son interface *upstream*. Il obéit dans son fonctionnement à des lois qui lui permettent de jouer le rôle de mandataire entre des hôtes MLD attachés à ses interfaces *downstream* et un routeur MLD situé relié à son interface *upstream*. Ces lois sont les suivantes :

1. Lorsque le *proxy* MLD est recensé par son routeur *upstream*, il répond par des rapports MLD au nom de tous les hôtes inscrits auprès de ses interfaces *downstream*. Ceci est rendu possible grâce aux informations d'abonnement aux groupes *multicast* que le *proxy* MLD collecte sur ses interfaces *downstream*.

2. Si l'un des hôtes *downstream* souscrit - en envoyant un rapport d'abonnement non sollicité - à une adresse *multicast* à laquelle aucun autre hôte n'est encore abonné auprès du *proxy* MLD, ce dernier envoie à son tour un rapport d'abonnement non sollicité associé à l'adresse en question via son interface *upstream*.
3. Si le dernier hôte abonné auprès du *proxy* MLD à une adresse *multicast* résilie son abonnement à cette adresse, le *proxy* envoie à son tour, via son interface *upstream*, un rapport MLDv1 de résiliation d'abonnement à l'adresse *multicast* all-routers (FF02 : :2), ou un rapport d'abonnement MLDv2 en mode "inclusion" avec une liste de sources vide à l'adresse all-MLDv2-routers (FF02 : :16).

Ainsi, les informations d'abonnements sont communiquées, grâce au protocole MLD, de fils à père sur l'arbre des *proxies* MLD. Chaque *proxy* maintient une base de données représentant l'agrégation de toutes les informations d'abonnements collectées sur ses différentes interfaces *downstream* [FEN04]. Cette base de données est utilisée par le *proxy* pour l'acheminement des paquets *multicast*.

4.2.1.4 Acheminement des paquets *multicast*

Un équipement *proxy* prend la décision d'acheminement des paquets *multicast* en suivant les règles suivantes :

1. Les paquets *multicast* reçus via l'interface *upstream* sont transmis via toute interface *downstream* vérifiant à la fois :
   (a) La base de données des abonnements contient une inscription au groupe *multicast* destination relative à cette interface.
   (b) L'équipement *proxy* est l'acheminer pour cette interface.
2. Les paquets *multicast* reçus via une interface *downstream* sont transmis sur :
   (a) Toute interface *downstream* autre que l'interface d'où provient le paquet vérifiant les conditions citées ci-dessus (1.a et 1.b).
   (b) L'interface *upstream* (sans conditions). Cette règle d'acheminement vise à permettre la transmission du trafic *multicast* émis par des sources internes dans le sens ascendant sur l'arbre des *proxies* MLD (vers la racine).

Afin d'éviter de prendre une décision d'acheminement pour chaque paquet arrivant, le *proxy* peut utiliser un cache qu'il met à jour à chaque fois que les informations intervenant dans cette décision changent (*cf.* conditions 1.a et 1.b).

Exemple : Considérons l'arbre de *proxies* MLD représenté par la figure 4.4. Soit une source *multicast* située à l'extérieur du réseau NEMO envoyant vers un groupe G, auquel est associé un arbre *multicast* dans l'Internet. Supposons qu'au début, aucun nœud dans le réseau mobile n'est abonné à G.

Quand le hôte MNN veut rejoindre G, il envoie un rapport d'abonnement MLD sur Lien5. Le *proxy* R4, agissant en tant que routeur MLD sur son interface *downstream* i3, intercepte ce message d'adhésion. Se rendant compte qu'il reçoit pour la première

fois un rapport d'abonnement à G provenant de i3, R4 stocke cette information dans sa base de données des abonnements et envoie, à son tour, un rapport d'abonnement MLD sur son interface *upstream* i1. En répétant ce même processus, la demande d'abonnement à G se propage sur l'arbre des *proxies* MLD de *proxy* "fils" à *proxy* "père" jusqu'à arriver au MR racine de l'arbre (R4→R2→MR). Configuré à son tour en *proxy* MLD, le MR aperçoit qu'il n'est pas encore abonné au groupe G. Il met à jour sa base de données des abonnements et envoie son propre rapport MLD pour G via son interface *upstream* i. Reçu par le routeur d'accès (AR1) sur le lien visité par le MR, ce message déclenche l'établissement d'une nouvelle branche de l'arbre *multicast* de G atteignant AR1.

A la réception de paquets *multicast* destinés à G, le MR vérifie les informations d'abonnements associées à chacune de ses interfaces *downstream* et achemine les paquets via les interfaces pour lesquelles le groupe G est mentionné (ici i2). Les autres routeurs dans le réseau mobile appliquent le même mécanisme de transmission et le paquet arrive enfin à MNN.

4.2.2 Problèmes de la solution basée sur le *proxying* MLD

4.2.2.1 Redondance du trafic liée à l'acheminement inconditionnel des paquets

La technique du MLD *proxying* autorise à plusieurs équipements *proxies* d'être connectés au même lien grâce au mécanisme d'élection d'un *proxy* acheminer. Si une telle configuration redondante représente pour l'administrateur du réseau un support facultatif de tolérance aux pannes, elle peut survenir à tout moment et de façon incontrôlée dans un réseau NEMO en conséquence de la variabilité de la topologie de celui-ci.

Tout risque de redondance du trafic n'est pourtant pas écarté par l'élection d'un seul *proxy* acheminer par lien. En effet, cette élection ne concerne que les interfaces *downstream* des *proxies* et laisse ainsi incontrôlée la redondance éventuelle liée au trafic provenant des interfaces *upstream*. Soit l'exemple suivant :

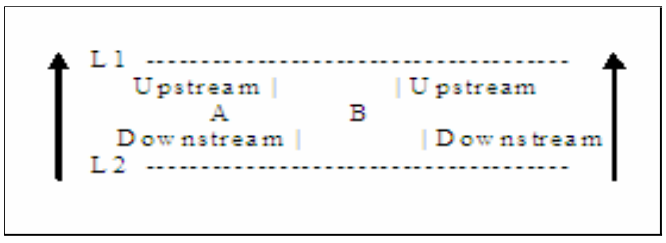


FIG. 4.6 – MLD *proxying* : Redondance du trafic

A et B sont configurés en *proxies* MLD avec chacun une interface *upstream* et une interface *downstream* comme représenté sur la figure 4.6. A un instant donné, seulement l'un des deux pourra acheminer le trafic *multicast* de L1 à L2 grâce au

mécanisme d'élection du recenseur/acheminеur, ce qui assure la transmission d'une seule copie de chaque paquet sur L2. Cependant, chaque paquet *multicast* est transmis en doublon dans le sens inverse (de L2 à L1) puisque A et B transmettent les paquets reçus via leurs interfaces *downstream* à leurs interfaces *upstream* de façon automatique et sans aucune condition.

4.2.2.2 Branches *multicast* externes inutiles

Lorsqu'un MNN adhère à une session (S,G), des messages d'abonnement MLD sont acheminés de façon automatique jusqu'au MR racine, qui à son tour émet un rapport d'abonnement MLD sur son interface externe, déclanchant ainsi la construction d'une branche *multicast* relative à (S,G) vers le MR racine. Si la source est elle-même interne au réseau NEMO, cette branche est construite vers le réseau mère (ancienne position du réseau de la source), en parallèle avec l'établissement d'un chemin optimal interne au réseau. Cette situation est illustrée par la figure 4.7.

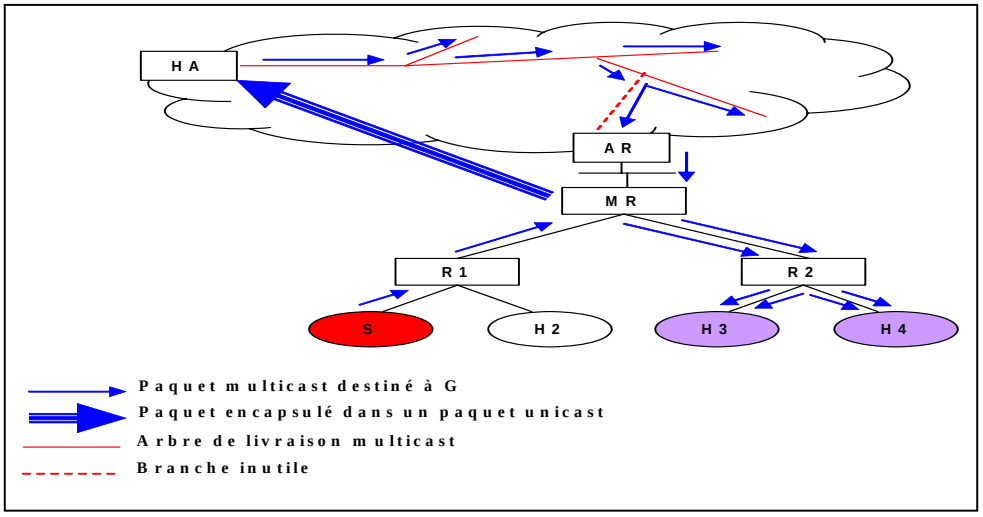


FIG. 4.7 – MLD *proxying* : branches multicast inutiles

L'adhésion des hôtes H3 et H4 à la session (S,G) conduit à la construction d'une branche de l'arbre *multicast* de (S,G) dans l'Internet desservant le routeur mobile MR. Cette même adhésion établit des états MLD permettant de joindre H3 et H4 à partir d'un chemin interne. Lorsque la source S émet un paquet *multicast* à destination de G, ce paquet est acheminé à H3 et H4 par deux chemins différents :

- Ce paquet est transmis sur le chemin direct inclus dans le réseau mobile (grâce aux règles d'acheminement définies par le *proxying* MLD)
- Ce paquet est conduit à travers le tunnel MR-HA vers l'agent mère (HA), qui le délivre sur l'arbre *multicast* associé à (S,G) . Or MR s'est enregistré à cet arbre. Le paquet revient alors à MR qui le délivre sur l'arbre des *proxies* MLD à H3 et H4.

4.2.2.3 Boucles de transmission

Considérons maintenant la configuration suivante :

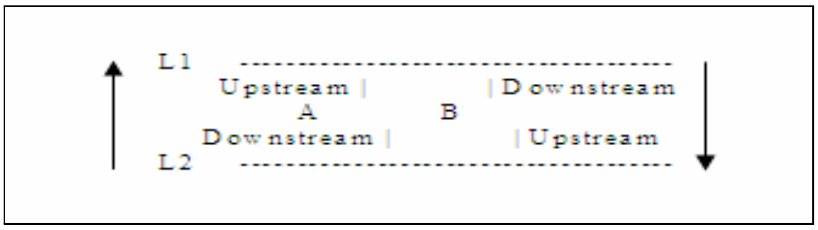


FIG. 4.8 – MLD *proxying* : boucle de transmission

B achemine inconditionnellement les paquets *multicast* de L1 à L2, et A achemine inconditionnellement les paquets *multicast* de L2 à L1 (*cf.* section 4.2.1.4). Il en résulte un problème beaucoup plus atroce que celui de la redondance du trafic puisque la transmission des paquets *multicast* entre les deux liens se fait selon une boucle infinie. Ce phénomène est appelé "boucle *upstream*".

Il est à noter que dans le cas normal, le problème de boucle *upstream* ne devrait pas subvenir puisque c'est une erreur de configuration que l'administrateur doit éviter. Cependant, cette erreur devient moins contrôlable dans le cas des réseaux mobiles dont la topologie change de façon dynamique. Dans notre exemple, si B est un routeur mobile qui vient s'attacher aux liens L1 et L2 comme décrit par la figure 4.8, il crée une boucle *upstream* qui n'était pas présente avant son arrivée.

4.3 Conclusion

L'intégration des deux composantes, mobilité et *multicast*, n'est pas triviale. Affectant les arbres de livraison *multicast* en partie ou en totalité, la mobilité IP des nœuds ou des réseaux exige des mécanismes spécifiques non fournis par les protocoles de routage *multicast* actuels.

Dans la première partie de ce chapitre, nous avons commencé par présenter différentes approches pour traiter le problème du *multicast* pour les nœuds IPv6 mobiles. Le protocole standard de mobilité normalisé par l'IETF dans les environnements IPv6 est Mobile IPv6. Nous avons présenté les deux mécanismes qu'il offre pour gérer les communications *multicast* : les tunnels bi-directionnels et l'enregistrement à distance. Chacune de ces deux approches présente des limites. Les tunnels bi-directionnels introduisent un routage triangulaire et génèrent un problème de "tunnels convergents". L'enregistrement à distance suppose la reconstruction partielle ou totale de l'arbre de diffusion. Il est donc non adapté au cas de noeuds mobiles qui se déplacent à une vitesse élevée, et non applicable au cas de sources SSM mobiles. Nous avons ensuite présenté d'autres alternatives aux solutions de l'IETF permettant de gérer les nœuds

multicast mobiles. Ces alternatives souffrent en général d'une difficulté de déploiement et ne passent pas à l'échelle avec la taille de l'Internet.

Dans une deuxième partie, nous nous sommes intéressés au problème du *multicast* pour les réseaux mobiles. Ce problème hérite de celui du *multicast* pour les nœuds mobiles, mais qui est nettement plus compliqué. En effet, outre l'enjeu de permettre aux nœuds à l'intérieur d'un réseau mobiles de garder la continuité de leurs sessions *multicast* en cours et d'établir de nouvelles sessions pendant le déplacement du réseau, une contrainte supplémentaire apparaît qu'est la nécessité de garder la mobilité du réseau transparente à ses nœuds, ceux-ci pouvant même être démunis de tout support de la mobilité. Ceci dit, le MR doit être chargé de gérer l'impact de la mobilité du réseau sur les communications *multicast* du reste des MNNs, ce qui peut être réalisé grâce aux approches classiques de gestion des nœuds mobiles, appliquées au niveau du MR au nom des MNNs. Ceci permet de réduire le problème initial à problème de livraison du trafic *multicast* dans le réseau NEMO.

Nous avons étudié dans ce chapitre une solution qui combine les approches classiques de gestion des nœuds *multicast* mobiles, à savoir l'enregistrement à distance et le *tunnelling* bidirectionnel, à la technique de *proxying* MLD, originellement proposée dans [FEN04] pour remplacer le déploiement d'un protocole de routage *multicast* dans un réseau de bordure à topologie simplifiée.

Le *proxying* MLD a l'avantage d'être facile à mettre en œuvre et d'offrir une transparence de la mobilité du réseau à ses nœuds *multicast*. Cependant, nous avons montré qu'il présente des inconvénients majeurs allant de la redondance du trafic et la construction inutile de branches *multicast* au risque de création de boucles de transmission lorsqu'il est déployé dans un réseau NEMO, dont la topologie peut changer à tout moment à cause de l'emboîtement de la mobilité.

Chapitre 5

Proposition d'un support *multicast* pour NEMO

Dans le but de fournir un service *multicast* aux réseaux NEMO, nous proposons de les munir d'un support de *multi-unicast* explicite. Se basant sur l'utilisation des chemins *unicast* pour la livraison du trafic multipoint, la technique de transmission Xcast a l'avantage d'éliminer tout risque de création de boucles de transmissions, même suite à des changements éventuels de la topologie.

Notre solution se base, plus précisément, sur le déploiement du protocole Xcast+6 à l'intérieur du réseau mobile. Nous avons choisi ce protocole car il a l'avantage de supporter les hôtes *multicast* standard, et d'offrir ainsi une transparence vis-à-vis de ces hôtes. Cependant, il nous a été nécessaire de prendre des mesures particulières pour pouvoir appliquer Xcast+6 au contexte spécifique des réseaux NEMO. En effet, loin de vouloir gérer les sessions multipoints de bout en bout par le protocole Xcast+6, notre but est de fournir un support pour les sessions *multicast* standard mettant en jeu des nœuds du réseau mobile. Nous utilisons alors le protocole Xcast+6 pour l'acheminement multipoint à l'intérieur des réseaux NEMO, tout en gardant inchangée la gestion de routage *multicast* dans le reste d'Internet. Outre l'utilisation du protocole Xcast+6 à l'intérieur du réseau mobile, nous avons recours aux approches d'enregistrement à distance et du tunneling bidirectionnel (*cf.* chapitre 4 section 4.1.1) au niveau du MR pour acheminer les paquets *multicast* respectivement vers et depuis le réseau mobile quand celui-ci n'est pas attaché à son réseau mère.

Pour des raisons de simplification, nous commençons par définir une solution de base qui ne tient pas compte de la question d'emboîtement. Nous étudions ensuite le comportement de notre solution de base dans le cas général plus compliqué où un nombre arbitraire de niveaux de mobilité sont emboîtés. Cette étude nous mène à proposer une optimisation de notre solution de base permettant de supporter de manière efficace l'emboîtement de la mobilité. Nous définissons ensuite les actions à effectuer après un déplacement du réseau NEMO. Enfin, nous évaluons notre solution en se basant sur différents critères.

5.1 Solution de base : en absence d'emboîtement

La solution de base que nous présentons ici suppose l'absence de toute forme d'emboîtement de la mobilité dans le réseau NEMO. Ce dernier est alors censé former une unité à topologie invariable. En particulier, tous les hôtes à l'intérieur du réseau sont des LFN.

5.1.1 Support pour les membres *multicast*

Nous nous intéressons en premier lieu aux récepteurs *multicast* situés à l'Intérieur du réseau NEMO. Notre but est alors de permettre aux MNNs de rejoindre des sessions *multicast* et d'en recevoir le trafic correspondant de façon normale, tout en gardant la mobilité du réseau transparente à ces nœuds.

Nous considérons pour le moment un réseau NEMO sans emboîtement, attaché soit à son lien mère soit à un lien étranger, et une session SSM (S,G).

5.1.1.1 Membres desservis par une source *multicast* externe

Supposons d'abord que la source S est externe, située dans l'infrastructure fixe d'Internet. Si un MNN veut adhérer à (S,G), il envoie un rapport d'abonnement MLD sur son lien local. Le routeur Xcast+6 jouant le rôle de DR de destination associé à cet MNN, que nous noterons par la suite DRdest, intercepte ce rapport MLD et envoie à S une demande d'enregistrement Xcast+6 contenant l'adresse de S, l'adresse du groupe G, et sa propre adresse [MYU01]. Cette demande n'est cependant pas acheminée jusqu'au réseau local de la source mais interceptée à mi-chemin par le MR. Ce dernier, jouant en quelque sorte le rôle d'un DR de source "anticipé", maintient une liste d'adresses des DRs de destination par session *multicast*. Il enregistre alors DRdest à la liste des DRs de destinations associée à (S,G). De plus, si c'est la première fois que le MR reçoit une demande d'enregistrement à (S,G), il émet un rapport d'abonnement MLD à (S,G) sur son lien externe. Notamment, le MR utilise son adresse CoA pour émettre des messages MLD s'il visite un lien étranger, et son HoA s'il est attaché à son lien mère. A la réception du rapport MLD du MR, le routeur *multicast* local (situé sur le lien mère ou étranger auquel le MR est attaché), déclenche la construction d'une branche *multicast* le reliant à l'arbre *multicast* de (S,G). Une fois cette procédure d'adhésion achevée, chaque paquet *multicast* émis par S à destination de G est transmis jusqu'au MR via l'arbre *multicast* associé à (S,G). Le MR transforme alors le paquet *multicast* qu'il reçoit en un paquet Xcast+6 par l'opération M2X [MYU01], et l'envoie vers la liste de DR de destinations enregistrés auprès de lui à (S,G).

Soit l'exemple de la figure 5.1 représentant un réseau NEMO non emboîté et une source *multicast* S externe envoyant vers un groupe G. Le routeur d'accès AR connecte le réseau NEMO à l'Internet. Nous supposons que AR est un routeur *multicast*, mais nous ne faisons aucune hypothèse sur la nature du lien auquel le MR est attaché (lien mère ou étranger).

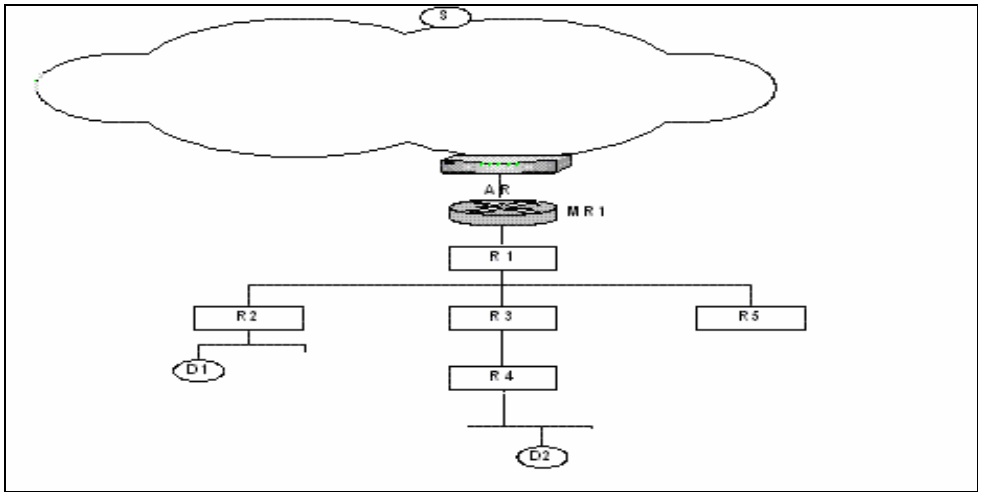


FIG. 5.1 – Support de base pour membres *multicast* (source externe)

Initialement, aucun MNN du réseau n'est membre de (S,G). Pour adhérer à (S,G), D1 envoie un rapport d'abonnement MLD à (S,G) sur son lien local. A la réception de ce rapport, R2 envoie à S une demande d'enregistrement Xcast+6 à (S,G). MR1 intercepte cette demande et, ne disposant pas encore d'une liste de DRs de destinations Xcast+6 pour (S,G), il crée cette liste, y insère l'adresse de R2 et envoie, sur son interface externe, un rapport d'abonnement MLD à (S,G) (en utilisant son adresse CoA s'il est sur un lien étranger). A la réception de ce rapport MLD, le routeur *multicast* AR déclenche la construction d'une branche de l'arbre *multicast* de (S,G) dans l'Internet vers le réseau NEMO.

Si par la suite D2 décide à son tour d'adhérer à (S,G), MR1 intercepte la demande d'enregistrement Xcast+6 envoyée par R4. MR1 se contente alors d'enregistrer R4 à la liste Xcast+6 de DR de destinations déjà créée lors de l'adhésion de D1.

Lorsqu'un paquet *multicast* est émis par S vers G, il est acheminé via l'arbre *multicast* de (S,G) jusqu'à MR1. Ce dernier le transforme, par le mécanisme M2X du protocole Xcast+6, en un paquet Xcast+6 destiné à la liste de DRs de destinations {R2,R4} associée à (S,G). Ce paquet est alors acheminé selon le protocole Xcast+6 jusqu'aux DRs de destinations R2 et R4, qui le retransforment en un paquet *multicast* standard par le mécanisme X2M [MYU01] et le délivrent respectivement à D1 et D2.

Lorsque le MR se déplace vers un nouveau lien de l'Internet, il se réinscrit (par envoi d'un rapport d'abonnement MLD) à toute session pour laquelle il maintient une liste Xcast+6 de DR de destinations, déclenchant ainsi la reconstruction d'une branche de l'arbre *multicast* correspondant vers sa nouvelle position. Ainsi, le trafic *multicast* continue à être acheminé vers les MNNs concernés sans que ces derniers se rendent compte du mouvement du réseau.

5.1.1.2 Membres desservis par une source *multicast* interne

Si la source S appartient à son tour au réseau NEMO auquel appartient le futur membre, la demande d'enregistrement Xcast+6 envoyée par le DR de destination est acheminée jusqu'au DR de S. Les paquets *multicast* émis par S seront alors transmis aux MNNs membres via le protocole Xcast+6 sur des chemins directs inclus dans le réseau NEMO.

Considérons l'exemple représenté par la figure 5.2. La source S, située à l'intérieur du réseau mobile, est entrain d'émettre du trafic *multicast* destiné à un groupe G.

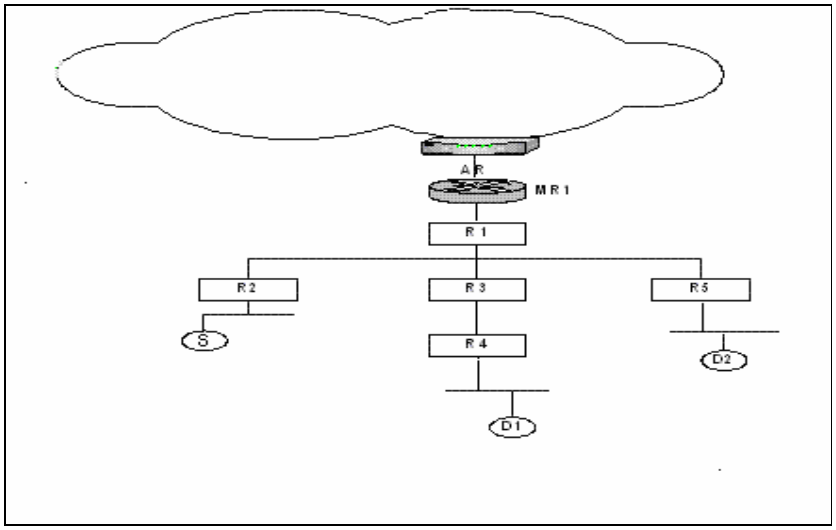


FIG. 5.2 – Support de base pour membres *multicast* (source interne)

Pour adhérer à (S,G), D1 émet un rapport d'abonnement MLD sur son lien local. A la réception de ce rapport, R4 envoie une demande d'enregistrement Xcast+6 à (S,G) vers S. De format *unicast* [MYU01], cette demande est alors transmise à R3, puis à R1, puis à R2. Ce dernier, jouant le rôle de DR de la source S, enregistre alors R4 à la liste Xcast+6 relative à (S,G). De même, si D2 envoie un rapport MLD pour s'abonner à (S,G), R5 s'enregistre auprès de R2 à la liste Xcast+6 associée à (S,G). Ainsi, R2 maintient la liste de DRs de destinations {R4,R5} relative à (S,G). Tout trafic *multicast* ultérieurement émis par S vers G est alors acheminé à D1 et D2 en utilisant le protocole Xcast+6 sur des chemins totalement compris dans le réseau mobile.

Il est à noter que les chemins établis dans ce scénario ne sont pas affectés par les déplacements du réseau. A titre d'exemple, D1 et D2 continuent à recevoir le trafic *multicast* relatif à la session (S,G) sur les mêmes chemins indépendamment du changement de la position du réseau NEMO. Aucun traitement n'est donc nécessaire lors d'un *handover* du MR.

5.1.2 Support pour les sources *multicast*

Considérons une source S située dans un réseau NEMO non emboîté et qui est entrain d'envoyer en vers un groupe *multicast* G. Notre but est alors d'assurer la livraison de ce trafic vers les différents membres concernés.

Nous venons de voir que les membres situés dans le même réseau NEMO que S peuvent être desservis en utilisant un chemin Xcast+6 direct (*cf.* section 5.1.1.2).

Afin d'atteindre le reste des membres de G (situés à l'extérieur du réseau NEMO), tout paquet *multicast* émis par S est régulièrement transmis au MR. Pour ce faire, le DR de S ajoute l'adresse du MR à la liste de destinations dans chaque paquet Xcast+6 qu'il génère par le mécanisme M2X à partir d'un paquet *multicast* émis par S. Le MR est alors chargé de retransformer le paquet Xcast+6 qu'il reçoit en un paquet *multicast* standard (mécanisme X2M) qu'il achemine via le tunnel MR-HA à son agent mère. Pour ce faire, le MR encapsule le paquet IPv6 *multicast* dans un paquet IPv6 *unicast* contenant son adresse CoA dans le champ adresse source et envoie ce paquet à son HA. Ce dernier désencapsule alors le paquet et le délivre sur l'arbre *multicast* associé à (S,G) dans l'Internet. Ainsi, le paquet est transmis aux diférents membres concernés. La figure 5.3 montre le mode de transmission utilisé pour l'acheminement du trafic multicast depuis une source située dans un réseau mobile vers des membres internes et externes à ce réseau.

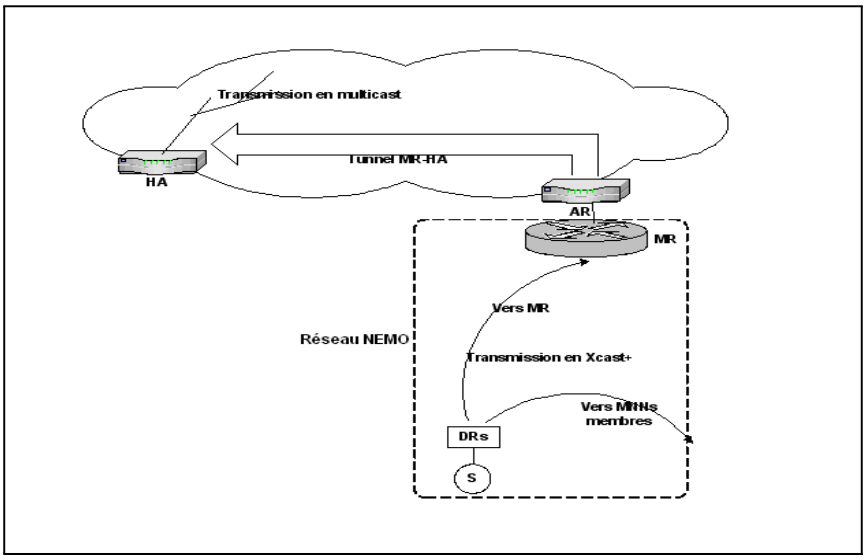


FIG. 5.3 – Support de base pour sources *multicast*

5.2 Impact de l'emboîtement sur la solution de base

Pour des raisons de simplification, nous avons commencé par définir une solution de base qui ne tient pas en compte la question d'emboîtement de la mobilité. Or dans un réseau NEMO emboîté à topologie arbitraire, il peut arriver qu'un MNN à n'importe quel niveau d'emboîtement ait besoin de participer à des sessions *multicast* en tant que source ou membre. Nous étudions ici l'impact du niveau d'emboîtement du MNN en question sur le comportement de notre solution de base. Dans tout ce qui suit, nous appelons NEMO élémentaire un réseau NEMO sans emboîtement de mobilité faisant partie d'un réseau NEMO plus grand.

5.2.1 Source externe envoyant vers des MNNs

Considérons un réseau NEMO emboîté et une source S externe à ce réseau entrain d'envoyer du trafic *multicast* vers un groupe G. Etant donné la transparence du niveau d'emboîtement des nœuds (fixes et mobiles) dans le réseau NEMO, un MNN déclenche la même procédure d'adhésion à (S,G) indépendamment de sa localisation dans la topologie du réseau NEMO.

Si le MNN en question est situé dans le NEMO racine, le résultat de cette procédure d'adhésion est identique à celui obtenu en absence d'emboîtement. En effet, le MR racine intercepte la demande d'enregistrement Xcast+6 envoyée par le DR de destination et effectue les traitements nécessaires.

Si, par contre, le MNN est situé à un niveau plus profond d'emboîtement, la demande d'enregistrement Xcast+6 passe d'abord par le MR du NEMO élémentaire auquel appartient le MNN. Ce dernier, n'étant même pas au courant de la mobilité du réseau étranger qu'il visite, se comporte exactement comme s'il était relié à l'infrastructure fixe de l'Internet. Ainsi, il enregistre le DR de destination du MNN concerné à la liste Xcast+6 relative à la session (S,G). De plus, si c'est pour la première fois qu'il reçoit une demande d'enregistrement Xcast+6 à (S,G), il émet un rapport d'abonnement MLD via son interface externe sur le lien visité, en utilisant sa CoA comme adresse source (*cf.* section 5.1.1.1). Comme ce lien fait partie du NEMO-père, le DR de destination Xcast+6 associé intercepte le rapport MLD et envoie en conséquence une demande d'enregistrement Xcast+6 à (S,G), qui sera interceptée et traitée par le MR-père. Ainsi, les enregistrements Xcast+6 se succèdent récursivement de sous MR à MR père jusqu'à arriver au MR racine. Ce dernier, comme nous l'avons déjà expliqué, rejoint l'arbre *multicast* associé à (S,G) dans l'Internet en envoyant un rapport MLD à (S,G) sur son lien visité. Il est à noter que si l'un des MRs intermédiaires intervenant dans la procédure d'adhésion s'aperçoit qu'il maintient déjà une liste Xcast+6 relative à (S,G), il n'envoie pas de rapport d'abonnement MLD, et la procédure récursive prend fin avant d'arriver au MR racine.

Lorsque un MR se déplace, il re-adhère à toutes les sessions *multicast* pour lesquelles il maintient une liste Xcast+6 de DR de destinations enregistrés (*cf.* section 5.1.1.1).

Aucune autre action n'est nécessaire puisque les chemins Xcast+6 établis entre ce MR et les MNNs appartenant à son arborescence (à n'importe quel niveau d'emboîtement) restent valides après ce déplacement.

Exemple :

Considérons l'exemple représenté par la figure 5.4.

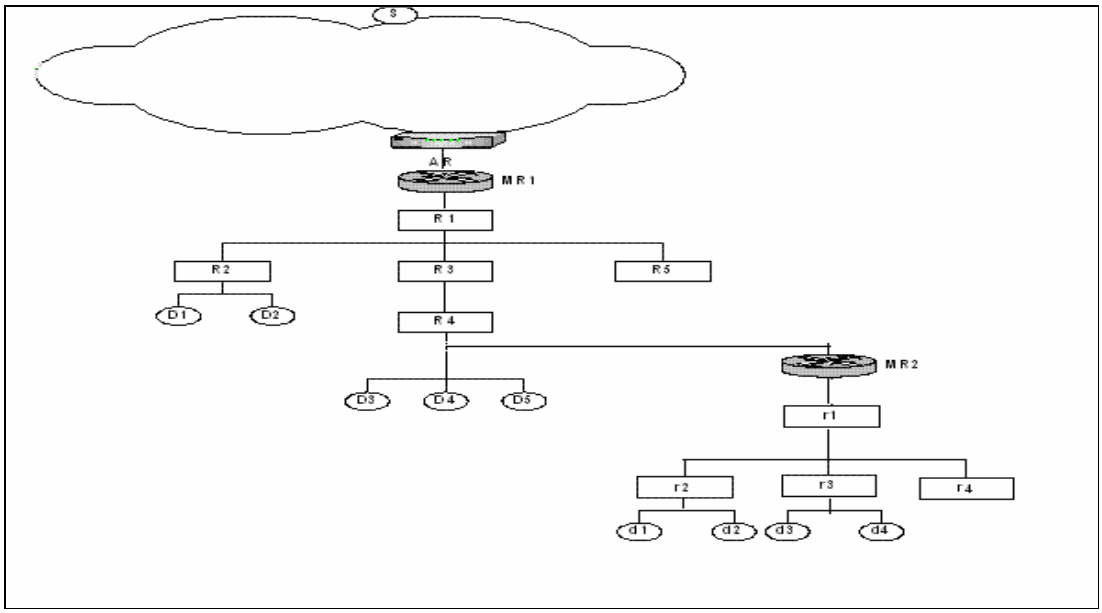


FIG. 5.4 – Support pour membres *multicast* dans un réseau emboîté (source externe)

Dans cet exemple, le réseau NEMO N2 vient s'attacher, via son routeur mobile MR2, au réseau NEMO N1 de routeur mobile MR1. L'ensemble forme alors un réseau NEMO emboîté N de racine MR1, connecté à l'Internet via le routeur d'accès AR. Nous supposons que ce dernier est un routeur *multicast* desservant le lien (mère ou étranger) auquel est attaché MR1. Les routeurs R1, .., R5 et r1, .., r4 sont des routeurs fixes appartenant respectivement à N1 et N2. S est une source externe envoyant du trafic *multicast* vers un groupe G. Nous supposons qu'initialement aucun MNN n'est membre de (S,G).

Si le hôte d1 veut adhérer à (S,G), il émet un rapport d'abonnement MLD correspondant sur son lien local. Son DR (r2) envoie alors une demande d'enregistrement Xcast+6 que MR2 intercepte est traite de la même manière que s'il était attaché à l'infrastructure fixe d'Internet : il crée une liste de DRs de destinations pour (S,G) et y insère r2, puis envoie un rapport d'abonnement MLD à (S,G) sur son interface externe en utilisant son adresse CoA. A la réception de ce rapport MLD, R4, jouant le rôle de DR de destination sur le lien visité par MR2, envoie à son tour à S une demande d'enregistrement Xcast+6 à (S,G). MR1 intercepte alors cette demande, crée une liste Xcast+6 pour (S,G), enregistre R4 à cette liste et émet un rapport d'abonnement MLD via son interface externe en utilisant son adresse CoA . A la réception

de ce rapport MLD, le routeur *multicast* AR déclenche la construction d'une branche de l'arbre *multicast* de (S,G) vers le réseau NEMO.

Supposons maintenant que les hôtes D1, D2, D3, D4, D5, d2, d3 et d4 aient aussi rejoint la session (S,G). Il est à noter que l'adhésion de ces hôtes se réduit à des simples mises à jour des listes Xcast+6 correspondantes, puisque MR1 et MR2 ont déjà rejoint la session (S,G) lors de l'adhésion de d1. Ainsi, au niveau de MR1, la liste résultante de DRs de destinations associée à (S,G) est {R2, R4}, alors que MR2 maintient la liste {r2,r3} pour cette même session.

Lorsqu'un paquet *multicast* est envoyé par S au groupe G, il arrive à MR1 via l'arbre de livraison *multicast* associé à (S,G). A la réception de ce paquet, MR1 le transforme en un paquet Xcast+6 destiné à la liste des DRs de destinations {R2,R4} qu'il transmet à R1. Ce dernier réplique alors ce paquet en deux copies, l'une destinée à {R2}, et l'autre à {R4}, qu'il transmet aux prochains nœuds respectifs R2 et R3 (*cf.* chapitre 2). Lorsque R2 reçoit le paquet Xcast+6 qui lui est destiné, il le transforme en un paquet *multicast* qu'il livre aux destinations finales D1 et D2.

Par ailleurs, R3 achemine la deuxième copie du paquet à R4. Ce dernier effectue à son tour un X2M et délivre le paquet *multicast* résultant sur son lien local. Ainsi, les membres D3, D4 et D5 sont desservis. De plus, à la réception de ce paquet *multicast*, MR2 se rend compte qu'il est associé à une session (S,G) pour laquelle il maintient la liste Xcast+6 de DRs de destinations {r1,r3}. MR2 agit de la même manière que MR1 : il transforme le paquet *multicast* qu'il vient de recevoir en un paquet Xcast+6 destiné à la liste {r2,r3}. Ce paquet est transmis jusqu'aux destinations finales d1, d2, d3 et d4 par le procédé d'acheminement de Xcast+6 [MYU01].

Si MR1 effectue un *handover*, il rejoint de nouveau la session (S,G). Comme les enregistrements Xcast+6 au niveau de MR2 et MR1 restent inchangés, le trafic associé à (S,G) est acheminé aux membres D1, ..., D5 et d1, ..., d4 de la même manière que lorsque MR1 était dans son ancienne position.

Si MR2 se déplace vers un nouveau lien de N1, son re-adhésion à (S,G) a pour résultat d'établir un nouveau chemin Xcast+6 de MR1 vers la nouvelle position de MR2. Comme ce dernier est toujours atteint par l'arbre de (S,G), et que MR2 détient encore la liste {r2,r3} relative à (S,G), le trafic correspondant continue à être acheminé vers d1, ..., d4 (les autres membres ne sont pas concernés par la mobilité de MR2).

Si MR2 se déplace vers un lien situé à l'extérieur du réseau NEMO emboîté, son re-adhésion à (S,G) déclenche la construction d'une branche *multicast* vers sa nouvelle position. Comme MR2 détient encore la liste Xcast+6 {r2,r3} pour la session (S,G), le trafic associé continue à atteindre les hôtes d1, ..., d4.

En conclusion, la solution de base permet de gérer efficacement l'envoi depuis une source *multicast* externe vers des nœuds situés dans un réseau NEMO à n'importe quel niveau d'emboîtement.

5.2.2 MNN envoyant vers des membres externes

Le trafic *multicast* envoyé par une source située dans un réseau NEMO à un niveau élevé d'emboîtement est tunnelé par le MR de même niveau d'emboîtement vers son

agent mère, qui le transmet sur l'arbre de livraison *multicast*. Notons, cependant, que les paquets subissent des encapsulations supplémentaires par les MRs du réseau intermédiaire qu'ils traversent, si le support de base de NEMO [RFC3963] est utilisé sans optimisation de routage.

5.2.3 Communication *multicast* intra-NEMO

Nous parlons de communication *multicast* intra-NEMO lorsqu'un MNN est membre d'une session *multicast* dont la source est à son tour un MNN du même réseau NEMO.

Nous avons vu que la communication intra-NEMO peut être facilement gérée par le protocole Xcast+6 si les MNNs membre et source font partie d'un même réseau NEMO non emboîté. Ceci reste vrai si ces deux MNNs font partie d'un même NEMO élémentaire appartenant à un réseau NEMO emboîté. En effet, la demande d'enregistrement Xcast+6 envoyée par le DR de destination est acheminée vers le DR de source sur le chemin *unicast* direct inclus dans le NEMO élémentaire en question.

Cependant, la situation plus générale où la source et le membre sont situés dans deux NEMO élémentaires différents faisant partie d'un même réseau NEMO emboîté s'avère moins évidente. En effet, si le support de base de NEMO [RFC3963] est utilisé sans optimisation de routage, le chemin *unicast* entre les DRs de destination et de source n'est pas un chemin direct inclus dans le réseau NEMO puisqu'il passe par les tunnels MR-HA correspondant. Nous nous proposons alors d'étudier le comportement de notre solution dans ce cas général de communication intra-NEMO, en discutant les différents scénarios possibles selon la position relative des réseaux NEMO élémentaires source et destination dans la topologie du réseau NEMO emboîté.

5.2.3.1 Réseau destination emboîté dans le réseau source

Dans cette section, nous étudions le scénario où le réseau destination est un sous-NEMO appartenant à la hiérarchie du réseau source à un niveau arbitraire d'emboîtement. Considérons pour ceci l'exemple de la figure 5.5.

Dans cet exemple, le réseau NEMO N3 de routeur mobile MR3 est attaché au réseau NEMO N2 de routeur mobile MR2, lui-même attaché au réseau NEMO N1 de routeur mobile MR1. L'ensemble forme un réseau NEMO N emboîté à trois niveaux de mobilité. Nous ne faisons aucune hypothèse sur le niveau de mobilité de MR1 (celui-ci peut alors être MR racine ou emboîté à son tour dans un autre réseau NEMO).

La source *multicast* S, située dans N1, émet du trafic destiné à un groupe *multicast* G. Pour adhérer à la session (S,G), le MNN D de N3 émet un rapport d'abonnement MLD sur son lien local. En conséquence, R31 envoie une demande d'enregistrement Xcast+6 à (S,G) destinée à S. En dépit du chemin *unicast* ordinaire que suivrait cette demande pour arriver à S (comprenant d'ailleurs les tunnels MR-HA des 3 MRs selon le support de base de NEMO [RFC3963]), elle passe nécessairement par MR3, point de connexion de N3 à la topologie extérieure. MR3 intercepte cette demande destinée à la source S qu'il considère comme étant externe, enregistre R31 à la liste Xcast+6

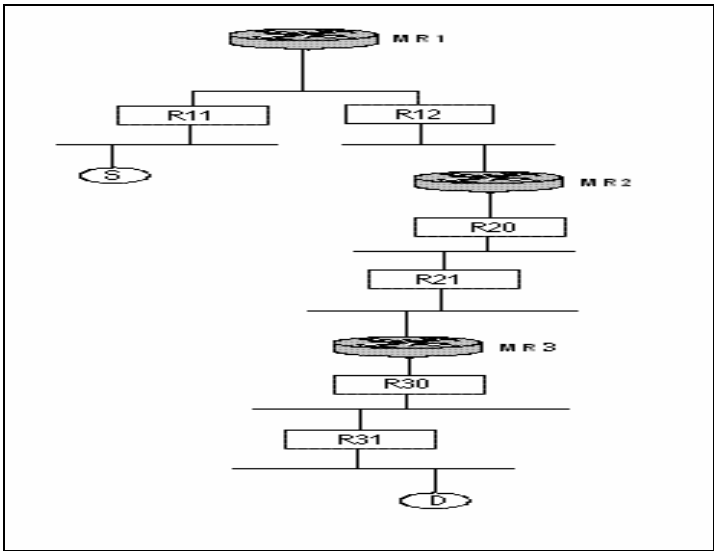


FIG. 5.5 – Réseau destination emboîté au réseau source

relative à (S,G) et émet (en utilisant sa CoA) un rapport d'abonnement MLD via son interface externe. La procédure se répète alors pour le sous-NEMO intermédiaire N2. En résultat, MR2 enregistre R21 à la liste Xcast+6 relative à (S,G) et émet un rapport d'abonnement MLD à (S,G) sur son lien externe.

Notons que jusqu'ici, l'opération d'adhésion suit la même procédure récursive déroulée dans le cas d'une source externe au réseau NEMO global. Ceci reste vrai quelque soit le niveau d'emboîtement du réseau NEMO destination. En effet, S est considérée comme étant une source externe par chaque sous-MR appartenant à l'arborescence emboîtée de racine MR2. Ainsi, au niveau de chaque sous-MR intermédiaire, nous avons un traitement identique à celui effectué par MR3 et MR2.

Cependant, S est reconnue comme étant interne au niveau du NEMO élémentaire N1. La demande d'enregistrement Xcast+6 émise par R12 (à la réception du rapport MLD de MR2) est alors transmise vers S sur le chemin direct inclus dans N1. R11 intercepte cette demande et enregistre R12 à la liste Xcast+6 correspondante. Notons que ce dernier enregistrement revient à une adhésion dans le même NEMO élémentaire.

A la fin de la procédure d'adhésion, on obtient le plan d'enregistrement Xcast+6 à (S,G) suivant :

- R31 (DR de destination de D) est enregistré auprès de MR3
- R21 (DR de destination de MR3) est enregistré auprès de MR2
- R12 (DR de destination de MR2) est enregistré auprès de R11

Ainsi, tout paquet *multicast* émis par S à destination de G est transmis en utilisant le protocole Xcast+6 sur 3 étapes : de S à MR2, de MR2 à MR3 et, de MR3 à D.

5.2.3.2 Réseau source emboîté dans le réseau destination

Voyons maintenant ce qui se passe si, à l'inverse du scénario précédent, le réseau source appartient à la hiérarchie du réseau destination. Pour ceci, considérons le même exemple précédent avec une simple modification consistant à échanger les emplacements de la source et du récepteur, ce qui donne la topologie représentée par la figure 5.6.

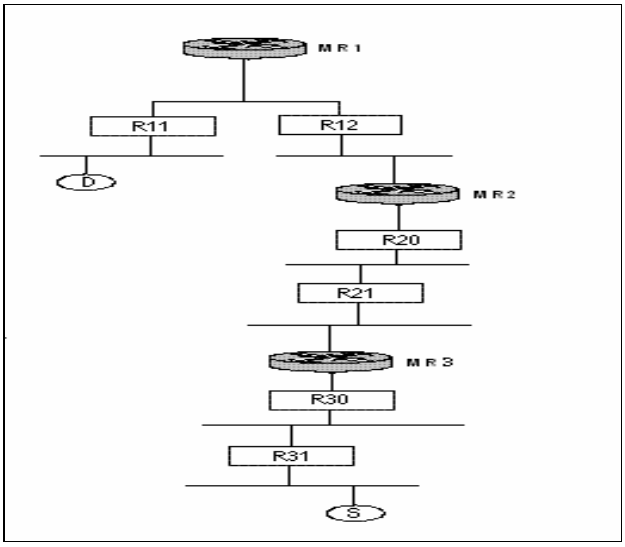


FIG. 5.6 – Réseau source emboîté au réseau destination

Lorsque D émet un rapport d'abonnement MLD pour adhérer à (S,G), son DR de destination R11 envoie à S une demande d'enregistrement Xcast+6 à (S,G). Or, si le support de base de NEMO est utilisé sans optimisation de routage, le chemin *unicast* vers tout nœud externe à N1 passe par MR1 [RFC3963]. MR1 intercepte alors la demande d'enregistrement, inscrit R11 à la liste Xcast+6 relative à (S,G) et émet à son tour un rapport d'abonnement MLD à (S,G) sur son interface externe. Si MR1 est un MR racine, ce rapport a pour effet de déclencher la construction d'une branche de l'arbre *multicast* de (S,G) vers le réseau NEMO emboîté. Sinon, des enregistrements Xcast+6 successifs ont lieu jusqu'au MR racine, qui déclenche de plus la construction d'une branche de l'arbre *multicast* de (S,G).

Ainsi, le trafic *multicast* émis par S vers G est d'abord tunnelé par MR3 jusqu'au réseau mère de N3, puis livré sur l'arbre de livraison *multicast* associé à (S,G) dans l'Internet jusqu'au MR racine, ensuite acheminé à D grâce au protocole Xcast+6. Un tel trafic doit alors passer par l'Internet avant d'arriver à D au lieu d'emprunter un chemin direct inclus dans le réseau mobile emboîté.

Cette non-optimalité des chemins est due à la transparence de la localisation des différents NEMO élémentaires du réseau NEMO emboîté les uns par rapport aux autres, ce qui empêche d'avoir une vision globale de la topologie du réseau mobile et d'établir des chemins directs. Rappelons que l'absence d'une vision globale de la

topologie emboîtée ne pose aucun problème si le réseau destination est emboîté au réseau source. En effet, nous avons montré dans la section précédente que les enregistrements récursifs allant du réseau destination au réseau source permettent d'établir un chemin Xcast+6 direct de la source vers la destination.

Dans la section qui suit, nous étudions le dernier scénario de communication *multicast* intra-NEMO correspondant à des sous-NEMO source et destination non emboîtés l'un dans l'autre.

5.2.3.3 Réseau source et réseau destination non liés par un lien d'emboîtement

Les réseaux NEMO source et destination peuvent appartenir à un même réseau NEMO sans, pourtant, être liés entre eux par un lien de parenté quelconque. Ce scénario est représenté par la figure 5.7.

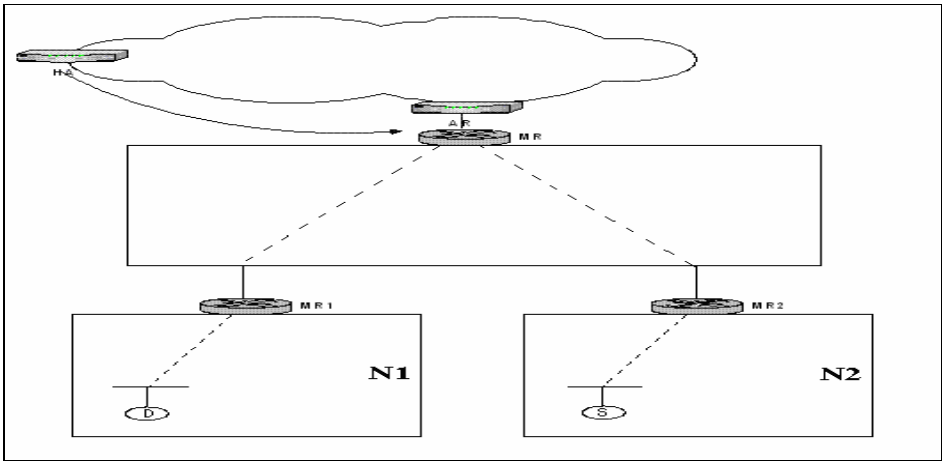


FIG. 5.7 – Réseaux source et destination non liées par un lien d'emboîtement

Sur cette figure, nous ne représentons pas tous les détails de la topologie du réseau NEMO emboîté. Les réseaux N1 et N2 sont deux NEMO élémentaires, les pointillés dans ces deux réseaux représentent alors des chemins constitués de routeurs fixes reliant respectivement D à MR1 et S à MR2. Par contre, les chemins reliant respectivement MR1 et MR2 au routeur mobile MR peuvent contenir d'autres niveaux d'emboîtement de la mobilité.

Supposons que D émet un rapport d'abonnement MLD pour adhérer à la session *multicast* (S,G). La procédure d'adhésion déclenchée a pour résultat l'établissement d'un chemin Xcast+6 relatif à (S,G) allant de MR à MR1, et la construction d'une branche de l'arbre *multicast* de (S,G) desservant MR. En effet, tous les routeurs intermédiaires entre MR1 et MR considèrent que S est une source externe et, par conséquent, reproduisent des rapports d'abonnement MLD sur leurs interfaces externes.

Nous nous retrouvons alors dans une situation similaire à celle d'un réseau source emboîté dans un réseau destination : le trafic *multicast* envoyé par S vers G doit être

transmis à l'agent mère de MR2 puis conduit sur l'arbre *multicast* de (S,G) avant de retourner vers N1.

5.2.3.4 Conclusion

L'étude du comportement de notre solution de base dans les différents scénarios de communication *multicast* intra-NEMO dans un réseau NEMO emboîté montre qu'elle présente un vrai problème d'optimalité des chemins dès que le NEMO élémentaire de destination n'appartient pas à la hiérarchie du NEMO élémentaire de source. En effet, le trafic passe inutilement par l'infrastructure d'Internet, au lieu d'emprunter un chemin direct dans le réseau NEMO.

Outre la longueur du chemin de livraison, cette transition inutile du trafic *multicast* par l'infrastructure d'Internet rend les communications *multicast* intra-NEMO vulnérables à la perte de la connectivité du réseau NEMO à l'Internet et aux pannes des routeurs externes traversés.

De plus, le passage par l'arbre *multicast* externe exige la mise à jour de cet arbre à chaque mouvement du réseau emboîté, ce qui augmente la charge du MR racine et la signalisation échangée et génère une rupture temporaire du service *multicast*.

Ainsi, il s'avère crucial de définir des mécanismes permettant d'optimiser notre solution de base afin qu'elle puisse gérer efficacement les communications intra-NEMO en fournissant des chemins de livraisons directs inclus dans le réseau NEMO.

5.3 Solution optimisée

L'absence d'une vision globale de la topologie du réseau NEMO emboîté est à l'origine du problème de non optimalité de chemin que nous avons discuté dans la section précédente.

Pour pallier à ce problème, nous avons recours à un mécanisme d'exploration permettant à chaque routeur mobile de localiser les sources *multicast* relativement à l'arborescence dont il est la racine.

5.3.1 Vue d'ensemble

Rappelons qu'un MR (racine ou non) qui reçoit, pour la première fois, une demande d'enregistrement Xcast+6 à une session à source externe, intercepte cette demande et émet un rapport d'abonnement MLD via son interface externe. Cependant, la source *multicast* en question risque d'être située dans l'arborescence du MR si cette arborescence présente un emboîtement, et le chemin établi est alors non optimal.

Afin de prévoir une telle situation, nous définissons une procédure d'exploration permettant au MR de vérifier l'emplacement relatif d'une source *multicast* par rapport à sa propre arborescence. Pour ce faire, le MR envoie à l'ensemble de ses sous-MRs directs un message d'interrogation, que nous appelons message *Find Request*, contenant l'adresse de la source S, du groupe G et celle du DR de destination voulant adhérer

à (S,G). Ainsi déclenchée, la procédure d'exploration se poursuit de façon récursive, grâce à des messages *Find Request* envoyés de MR père à sous-MRs. Si l'un des MRs sur l'arborescence explorée trouve que la source est bien joignable via l'une de ses interfaces internes, un chemin Xcast+6 direct est établi entre les DRs de source et de destination via des enregistrements Xcast+6 adéquats. Sinon, le MR qui a déclenché la procédure d'exploration est informé que la source n'est pas dans son arborescence, auquel cas il émet un rapport d'abonnement MLD sur son interface externe.

Chaque routeur mobile maintient un cache contenant les informations de localisation de sources qu'il a pu collectées lors des explorations auxquelles il a participé. Ce cache lui permet d'optimiser les enregistrements ultérieurs aux sources déjà recherchées.

5.3.2 Informations de topologie

Pour explorer son arborescence, un MR utilise son adresse HoA pour envoyer des messages de contrôle *Find Request* à l'ensemble de ses sous-MRs directs. Afin de minimiser la consommation de la bande passante liée à l'envoi de tels messages, nous utilisons une transmission en Xcast+6. Par ailleurs, ceci permet de limiter les informations de topologie que doit acquérir un MR à la liste des liens visités par ses sous-MRs directs.

Pour ce faire, nous commençons par définir un groupe *multicast* spécifique d'adresse *All_MRs*, auquel appartient constamment tout routeur mobile. Ce groupe spécial servira d'identificateur de canal dans les paquets Xcast+6 d'interrogations envoyés par un MR à ses sous-MRs directs.

Un MR ne doit émettre aucun rapport MLD pour adhérer au groupe *All_MRs*, puisqu'il est incapable de déterminer s'il est ou pas attaché à un autre réseau NEMO. Chaque DR de destination dans un réseau NEMO détecte la présence éventuelle de sous-MRs sur le lien dont il a la charge, grâce à la découverte de voisinage [RFC2461], et s'enregistre au nom de ces sous-MRs à la liste Xcast+6 (MR père, *All_MRs*) auprès du MR père. Pour ce faire, le DR chargé d'un lien envoie au MR père une demande d'enregistrement Xcast+6 relative à la session (MR père, *All_MRs*), à la détection d'un premier routeur mobile sur ce lien. Si, par contre, tous les routeurs mobiles quittent le lien, le DR envoie une demande de désabonnement Xcast+6 correspondante au MR père.

La demande d'enregistrement envoyée par le DR est acheminée jusqu'au MR père en question. Ce dernier détient par lui-même une liste Xcast+6 des DRs de destinations chargés des liens visités par ses sous-MRs directs. Notons que la taille de cette liste ne peut pas dépasser le nombre de liens fixes du réseau NEMO père, puisque chaque lien est contrôlé par un DR de destination Xcast+6. Notons aussi que le MR maintenant par lui-même la liste Xcast+6 relative à une session dont il est la source, ce qui représente une forme de transgression de la spécification de Xcast+6 selon laquelle une source est différente de son DR. Cette fusion de rôles n'introduit cependant aucune complexité supplémentaires au déploiement de Xcast+6 : il suffit de configurer le MR pour jouer le rôle de son propre DR, en lui confiant la gestion des demandes

d'enregistrements qui lui sont destinées et l'encapsulation de ses paquets *multicast* sortants relatifs à la session (MR,*All_MRs*) dans des paquets Xcast+6 avant de les envoyer.

5.3.3 Format d'un paquet d'interrogation

Un message *Find Request* contient les mêmes champs qu'un message de demande d'enregistrement Xcast+6 (*Registration Request*), mais il est contenu dans un en-tête d'extension de destination (*Destination Header*) d'un paquet Xcast+6 que nous appelons paquet d'interrogation .

Un paquet Xcast+6 d'interrogation envoyé par le MR a donc le format général suivant :

[En-tête IPv6 | en-tête de routage {Xcast+6 }| En-tête de destination {Find Request}]

L'entête IPv6 du paquet contient l'adresse mère (HoA) du MR qui a envoyé le message *Find Request* et l'adresse *All_Xcast_Routers* respectivement dans les champs adresse source et adresse destination (*cf.* Annexe B). Cet en-tête est suivi par un en-tête d'extension de routage (*Routing Header*) contenant les informations de routage Xcast+6. Le champ identificateur de canal de ce deuxième en-tête contient l'adresse *All_MRs*, alors que la liste de destinations contient les DRs de destinations à interroger. Enfin, l'en-tête d'extension de destination (*Destination Header*) contient l'adresse de la source *multicast* S et du groupe G relatifs à la session sur laquelle porte l'interrogation, et l'adresse du DR de destination à enregistrer (*cf.* figure 5.8).

Notons que le transport du message *Find Request* dans un en-tête de destination empêche le paquet Xcast+6 d'interrogation de contenir l'en-tête de destination optionnel de Xcast+6 relatif à liste de ports.

Un message *Find Request* est contenu dans un en-tête IPv6 d'extension de destination dont le format est représenté par la figure 5.8.

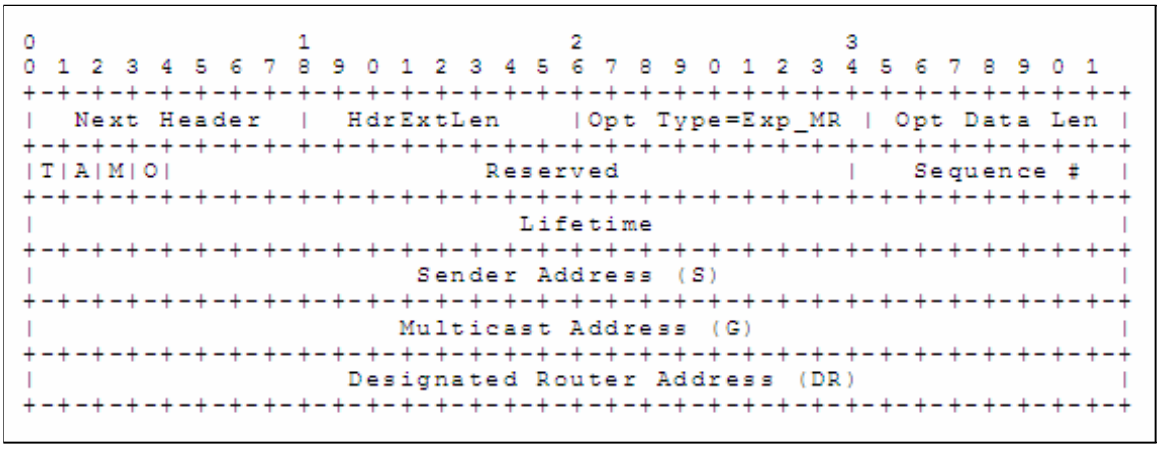


FIG. 5.8 – Format d'un message *Find Request*

Le type d'option "Exp_MR" indique que c'est un message *Find Request* et doit être assigné par IANA, ses trois premiers bits doivent être 010 pour indiquer que le paquet doit être rejeté si l'option est inconnue, et que l'option ne peut pas être changée en cours de route.

Les autres champs sont les mêmes qu'un message de demande d'enregistrement Xcast+6 (*cf.* annexe C).

Le bit T est mis à 0 pour indiquer qu'il s'agit d'une demande d'enregistrement à (S,G), et le bit A à 1 pour forcer l'envoi d'un acquittement. Un message *Find Request* est acquitté par un message de type *Registration Reply* défini par Xcast+6 (*cf.* Annexe C).

5.3.4 Algorithme d'exploration

5.3.4.1 Description de l'algorithme

Lorsqu'un MR reçoit, pour la première fois, une demande d'enregistrement Xcast+6 à une session (S,G) provenant d'un routeur désigné DRdest, et qu'il trouve que le chemin *unicast* vers S passe par son interface externe, il n'émet pas de façon automatique un rapport d'abonnement MLD sur son lien externe mais commence par déclencher une procédure d'exploration de son arborescence à la recherche de S.

Notons par *Sub_MR_List* d'un routeur mobile MR la liste Xcast+6 associée à la session (MR,*All_MRs*).

La procédure déroulée au niveau du MR qui a déclenché l'exploration est la suivante :

1. Si la liste *Sub_MR_List* du MR est vide (autrement dit le MR n'admet pas de sous-MRs), la procédure prend fin avec résultat : S source externe.
2. Sinon (*Sub_MR_List* non vide), le MR effectue les actions suivantes :
   (a) Il envoie un message *Find Request* contenant l'adresse de S, de G et celle de DRdest, encapsulé dans un paquet Xcast+6 d'interrogation d'adresse source la HoA du MR, destiné à la liste *Sub_MR_List* et d'identificateur de canal égal à *All_MRs* (*cf.* section 5.3.3).
   (b) Il se met en état d'attente des messages d'acquittement d'enregistrement Xcast+6 (*Registration Reply*) correspondants provenant des DRs appartenant à *Sub_MR_List*. Cette attente s'arrête lorsque l'une des conditions suivantes est satisfaite :
      i. Réception d'un message d'acquittement *Registration Reply* positif, en réponse au message *Find Request* envoyé. Le MR ignore alors toutes les réponses ultérieures à son *Find Request*.
      ii. Réception des réponses *Registration Reply* de tous les DRs appartenant à la liste *Sub_MR_List*.

(c) Si un message d'acquittement *Registration Reply* positif a été reçu en réponse au message *Find Request* envoyé, le résultat de la procédure est : S source interne, accessible via le routeur qui a envoyé le *Registration Reply* positif.

(d) Sinon (Tous les acquittements reçus sont négatifs), le résultat est le suivant : S est une source externe.

Le MR agit alors selon le résultat de la procédure d'exploration :

1. Si la source S s'est avérée interne, le MR ne fait rien puisque la procédure d'exploration est censée effectuer par elle-même les enregistrements nécessaires.
2. Si, au contraire, S s'est avérée source externe, le MR enregistre DRdest à une nouvelle liste Xcast+6 associée à (S,G) et émet sur son lien externe un rapport d'abonnement MLD relatif à (S,G).

Lorsqu'un routeur désigné DRi appartenant à la liste *Sub_MR_List* reçoit le paquet Xcast+6 d'interrogation contenant le message *Find Request* du MR, il effectue le suivant :

1. Il transforme le paquet reçu en un paquet *multicast* standard destiné au groupe *All_MRs* et contenant le message *Find Request* par l'opération X2M, puis livre le paquet résultant sur le(s) lien(s) qu'il dessert et qui hébergent des routeurs mobiles (puisque tout routeur mobile appartient au groupe *All_MRs*), conformément au protocole Xcast+6 [MYU01].
2. Il reste en écoute des acquittements *Registration Reply* envoyés par les sous-MRs sur son lien local vers le MR père. Il intercepte les messages d'acquittement négatifs et ne les laisse pas poursuivre leur chemin. DRi traite ces acquittements comme suit :

   (a) Si un acquittement *Registration Reply* positif correspondant est reçu, DRi laisse ce message poursuivre son chemin.

   (b) Si tous les MRs attachés au lien ont répondu par des messages *Registration Reply* négatifs, il envoie au MR père un unique *Registration Reply* Négatif.

Voyons maintenant la procédure déroulée par chaque routeur mobile situé sur l'arborescence explorée. Lorsqu'un DR délivre le paquet *multicast* contenant le message *Find Request* sur son lien local, tous les MRs attachés à ce lien reçoivent ce paquet, destiné au groupe *All_MRs* auquel ils appartiennent. Chaque sous-MR (MRi) déroule alors la procédure suivante :

1. Si MRi trouve que le chemin vers S passe par une interface interne, alors il enregistre le DR de destination indiqué par le message *Find Request* qu'il vient de recevoir à une nouvelle liste Xcast+6 relative à (S,G) et envoie, en utilisant son adresse CoA comme adresse source, un acquittement *Registration Reply* positif destiné à l'adresse HoA de son MR père (puisque c'est l'adresse source du message *Find Request* reçu). De plus, si le message *Find Request* reçu contient

un champ adresse DR de destination dont la valeur est différente de celle du champ adresse source, MRi envoie un deuxième *Registration Reply* positif vers ce DR.

2. Sinon (le chemin vers S passe via l'interface externe de MRi), alors MRi déroule la même procédure d'exploration pour (S,G) que le MR déclencheur de l'exploration, avec la seule différence que le message *Find Request* envoyé par MRi contient l'adresse HoA de MRi lui-même dans le champ adresse DR de destination. Il utilise aussi son adresse HOA comme adresse source du paquet d'interrogation et du message *Find Request* (*cf.* section 5.3.3). Selon le résultat de la procédure d'exploration déclenchée par MRi, ce dernier agit comme suit :

   (a) Si le résultat est "S source interne" alors MRi procède comme dans le cas 1.

   (b) Sinon (le résultat est "S source externe"), MRi envoie un acquittement *Registration Reply* négatif à son MR père (rappelons que ce message est intercepté par le DR de destination sur le lien externe visité de MRi, qui est en état d'attente des réponses des MRs sur le lien).

5.3.4.2 Tables de cache Xcast+6

Un MR maintient deux tables de cache Xcast+6 différentes :
- La table de cache Xcast+6 à sources externes : cette table contient les listes Xcast+6 de DRs de destination relatives à des sessions à sources externes à son arborescence.
- La table de cache Xcast+6 à sources internes : cette table contient les listes Xcast+6 des DRs de destinations relatives à des sessions à sources internes à son arborescence.

Cette distinction entre les listes Xcast+6 basée sur l'emplacement de la source est nécessaire pour déterminer les traitements à effectuer lors d'un *handover* du MR (*cf.* section 5.4).

5.3.4.3 Cache de localisation des sources

Chaque routeur (MR ou DR) qui a participé à une exploration relative à une source donnée maintient l'information de localisation qu'il a pu acquérir grâce à cette exploration dans un cache spécifique.

Cette information est notamment relative à chaque routeur (mobile ou fixe), elle lui indique si la source est attachée ou pas à son arborescence, et elle est de la forme :

"S source externe" ou "S source interne accessible via le routeur MRi", où MRi désigne un sous-MR.

Avant d'entamer une exploration de son arborescence à la recherche d'une source qui est externe selon sa table de routage *unicast*, un MR vérifie d'abord son cache de localisation des sources, afin d'éviter de rechercher deux fois la même source.

De même, à la réception d'un paquet Xcast+6 d'interrogation, un DR vérifie d'abord son cache de localisation des sources. Si ce cache contient une information concernant la source recherchée, il répond directement sans délivrer l'interrogation aux MRs sur le lien.

5.3.4.4 Sollicitations parallèles de la même source

Si un routeur (MR ou DR), entrain d'effectuer une exploration relative à une session (S,G1), reçoit une nouvelle sollicitation (demande d'enregistrement Xcast+6 ou message *Find Request*) nécessitant la localisation de S (le groupe G2 de la nouvelle session en question pouvant être différent de G), il ne relance pas de nouvelle exploration. Une fois l'exploration pour (S,G1) terminée, il utilise son résultat pour répondre à la demande l'adhésion à (S,G2).

5.3.5 Exemple

Nous reprenons dans la figure 5.9 l'exemple d'un réseau source emboîté dans un réseau destination, et nous expliquons la nouvelle procédure d'adhésion de D à (S,G) utilisant l'algorithme d'exploration pour l'optimisation des chemins.

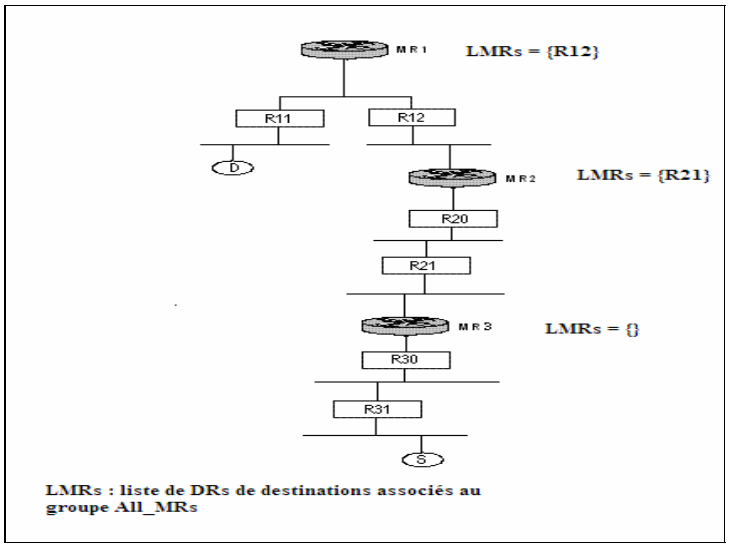


FIG. 5.9 – Exemple de déroulement de l'algorithme d'exploration

Notons d'abord que pour cette topologie, MR1 et MR2 détiennent respectivement les listes Xcast+6 {R12} relative à la session (MR1, *All_MRs*) et {R21} relative à la session (MR2, *All_MRs*).

Si le hôte D envoie un rapport d'abonnement MLD pour adhérer à (S,G), son DR, R11, envoie à S une demande d'enregistrement Xcast+6 à la session (S,G), que MR1 intercepte. Nous supposons que MR1 reçoit pour la première fois une demande

d'enregistrement Xcast+6 à (S,G). Il consulte alors sa table de routage *unicast* et trouve que S est une source externe (accessible via un chemin *unicast* passant par l'interface externe de MR1). Cependant, au lieu d'émettre de façon automatique un rapport d'abonnement MLD sur son lien visité, MR1 commence par vérifier si S appartient en réalité à son arborescence emboîtée, auquel cas un chemin direct contenu dans le réseau NEMO existe. Pour ce faire, MR1 consulte d'abord son cache de localisation des sources. Nous supposons qu'il n'y trouve aucune information concernant S. Il déclenche alors une procédure d'exploration en envoyant un message *Find Request* contenant l'adresse de S, l'adresse du groupe G et celle de R11, vers sa liste Xcast+6 de DRs de destinations {R12}.

Lorsque R12 reçoit le paquet Xcast+6 d'interrogation contenant le message *Find Request*, il le transforme, par le mécanisme X2M, en un paquet *multicast* standard destiné au groupe *All_MRs* [MYU01], qu'il émet sur le lien local hébergeant MR2, et se met en état d'attente. A la réception de ce paquet, MR2, appartenant au groupe *All_MRs*, consulte à son tour sa table de routage *unicast* et trouve que S est aussi accessible via son interface externe. Après consultation sans résultat de son cache de localisation des sources, MR2 envoie à son tour un message *Find Request*, dans un paquet Xcast+6 destiné à la liste Xcast+6 {R21}. MR2 utilise son adresse HoA pour l'envoi de ce message *Find Request*, qui contient aussi cette même adresse dans le champ DR de destination.

Lorsque MR3 reçoit le message *Find Request* de MR2, il se rend compte que la source *multicast* recherchée S est interne. MR3 enregistre alors l'adresse HoA de MR2 à sa liste Xcast+6 de DRs de destinations relative à (S,G), et lui envoie un message Xcast+6 d'acquittement d'enregistrement (*Registration Reply*) positif. A la réception de cet acquittement, MR2 enregistre à son tour R11 à (S,G) et envoie à MR1 et à R11 chacun un message *Registration Reply* positif contenant R11 dans le champs DR de destination (puisque le message *Find Request* provenant de MR1 contient l'adresse de R11 dans le champ DR de destination).

Le résultat final de la procédure d'adhésion de D à la session (S,G) est donc le suivant :

- MR2 est enregistré (avec son adresse HoA) à la liste Xcast+6 relative à (S,G) de MR3
- R11 est enregistré à la liste Xcast+6 relative à (S,G) de MR2

5.4 Gestion des mouvements du réseau

Un *handover* du MR racine affecte l'ensemble du réseau NEMO emboîté, alors qu'un *handover* d'un sous-MR affecte la portion du réseau NEMO emboîté desservie par ce sous-MR (c'est-à-dire l'arborescence située derrière ce sous-MR), elle même constituant un réseau NEMO emboîté ou élémentaire. Dans les deux cas, un chemin de livraison déjà établi dans peut devenir non valide pour l'une des deux raisons suivantes :

1. L'arbre *multicast* ne dessert plus le réseau NEMO ayant effectué le *handover* dans sa nouvelle position.
2. Les enregistrements Xcast+6 relatifs à la partie interne de ce chemin ne sont plus adéquats.

Il est à noter que tout chemin totalement inclus dans le réseau NEMO qui s'est déplacé reste valide. Par ailleurs, dès que le MR obtient sa nouvelle adresse CoA, le trafic *multicast* issu des sources impliquées dans le mouvement peut être conduit à travers le tunnel MR-HA et livré sur les arbres de livraison *multicast* associés. Cependant, un tel trafic risque de ne plus être délivré aux membres *multicast* situés dans d'autres parties du réseau NEMO emboîté auquel était attaché le réseau NEMO en question avant son déplacement. De plus, les récepteurs *multicast* à l'intérieur du réseau NEMO déplacé ne peuvent plus recevoir le trafic provenant d'une source externe à ce réseau, même si cette source fait partie de réseau NEMO emboîté d'origine.

Afin de garder la continuité des sessions *multicast* en cours mettant en jeu des membres ou sources *multicast* à l'intérieur d'un réseau NEMO, un MR doit effectuer deux types d'actions après chacun de ses *handover*s : une re-inscription aux sessions *multicast* et une notification du déplacement des sources.

5.4.1 La re-inscription aux sessions *multicast*

Pour chaque liste de DRs de destinations appartenant à sa table de cache Xcast+6 à sources externes, le MR ayant effectué le *handover* envoie un rapport d'abonnement MLD sur son interface externe pour re-adhérer à la session associée.

S'il s'agit d'un sous-MR qui s'est déplacé en restant dans le même réseau emboîté, cette re-adhésion a pour résultat la mise à jour des chemins décris par des enregistrements Xcast+6 vers la nouvelle position du MR. Dans le cas d'une source totalement externe au réseau NEMO emboîté, la branche d'arbre *multicast* reste valide puisque elle dessert toujours le MR racine, ainsi aucune reconstruction d'arbre n'a lieu.

S'il s'agit d'un sous-MR qui a complètement quitté le réseau NEMO emboîté, ou d'un MR racine qui s'est déplacé vers un nouveau lien de l'Internet, cette re-adhésion permet d'établir des nouveaux chemins de livraison du trafic *multicast*. Si le nouveau lien visité vers le MR fait partie de l'infrastructure fixe de l'Internet, de nouvelles branches *multicast* sont construites dans l'infrastructure de l'Internet pour desservir ce lien. Si, par contre, le lien visité fait partie d'un nouveau réseau NEMO, des branches *multicast* sont construites pour desservir le MR racine du réseau NEMO visité, et des enregistrements Xcast+6 adéquats sont établis pour permettre la livraison du trafic *multicast* du MR racine vers le MR qui a subit le *handover*.

5.4.2 La notification des déplacements des sources

Après un *handover*, le MR notifie chaque DR de destination enregistré auprès de lui à une liste Xcast+6 à source interne du déplacement de la source (induit par

le *handover* du MR). Pour ce faire, il utilise la sous option "*SSM source handover notification*" définie dans [JEL02a] de l'option *IPv6 Binding Destination Option* (*cf.* annexe D). Notons, cependant, que le champ *New Source Address* du message de notification contient l'adresse principale de la source, et non une nouvelle CoA (d'ailleurs, la source n'acquiert pas de CoA après déplacement du MR).

Un seul message de notification est envoyé pour un DR donné concernant une source donnée, même si ce DR est enregistré auprès du MR à plusieurs sessions *multicast* ayant la même source. Si le DR notifié est lui-même un routeur mobile intermédiaire, il envoie à son tour une notification à chacun des DRs enregistrés auprès de lui à des listes Xcast+6 associées à la source S. Ceci permet de faire arriver les notifications aux DRs de destinations finales intéressées par des sessions de source S. Chaque DR de destination finale notifié re-adhère à toutes les sessions de sources S pour les quelles il y a encore des nœuds intéressés sur le(s) lien(s) dont il a la charge, notamment en envoyant les demandes d'enregistrement Xcast+6 adéquats. Ceci permet de re-établir des chemins de livraison valides allant des sources *multicast* appartenant au réseau qui s'est déplacé vers les récepteurs *multicast* situés dans la partie du réseau NEMO emboîté non affectée par le *handover* du MR en question.

Revenons à l'exemple de la section 5.3.5. Supposons que MR3 se déplace vers un nouveau lien visité. Nous ne faisons aucune hypothèse sur l'emplacement de ce lien, qui peut alors être situé à l'intérieur ou à l'extérieur du réseau NEMO emboîté. Puisque MR2 est enregistré (par son adresse principale HoA) auprès de MR3 à la liste Xcast+6 relative à la session (S,G), et que cette liste est à source interne, MR3 envoie un message de notification contenant l'adresse de S vers MR2 (destiné à la HoA de MR2). A la réception de ce message, MR2 notifie à son tour le routeur R11 par un message similaire. Ainsi, R11, informé du déplacement de S, relance la procédure d'adhésion à (S,G) en envoyant une demande d'enregistrement Xcast+6 à cette session.

5.5 Evaluation de la proposition

Afin d'évaluer notre solution, nous nous basons sur les critères suivants : nature de la livraison, qualité des chemins établis, absence de boucles de transmission, transparence vis-à-vis des MNNs, fonctionnement en mode déconnecté, mobilité globale dans l'Internet et compatibilité avec le modèle ASM.

5.5.1 Nature de la livraison

Basée sur l'utilisation du protocole Xcast+6 à l'intérieur du réseau NEMO, notre solution permet d'établir des chemins de livraison multipoint. Ainsi, les envois répétitifs sur un même lien du même paquet sont évités, sans pour autant inonder tout le réseau par le trafic *multicast*.

5.5.2 Qualité des chemins

Notre solution permet d'établir des chemins optimaux vers les membres situés dans le réseau NEMO, grâce à l'utilisation des tables de routage *unicast* par le protocole Xcast+6 [MYU01] à l'intérieur du réseau mobile, et à la re-construction de l'arbre s'il s'agit d'une source externe. L'algorithme d'exploration permet d'assurer l'optimalité des chemins lors de l'établissement de communications *multicast* intra-NEMO en présence d'emboîtement.

Cependant, dans le cas d'un MNN source *multicast* envoyant du trafic *multicast* vers l'extérieur, plusieurs encapsulations ont lieu si la source est à niveau profond d'emboîtement, ce qui engendre un chemin non optimal. Mais il est à noter que ce problème est lié au tunneling et est inévitable si le support de base de NEMO est utilisé sans optimisation de routage. Or des solutions ont été déjà définies pour l'optimisation du tunneling pour NEMO. L'utilisation de ces solutions permet de résoudre le problème.

5.5.3 Absence de boucles de transmission

Grâce à l'utilisation des chemins *unicast*, l'absence de risque de création de boucles de transmission est automatiquement assurée par le protocole de routage *unicast* sous-jacent.

5.5.4 Transparence vis-à-vis des MNNs

Les hôtes *multicast* à l'intérieur du réseau mobile sont des hôtes IPv6 standard, ils ne sont censés implémenter aucune fonctionnalité additionnelle. En effet, la mobilité du réseau est gérée au niveau du MR.

5.5.5 Fonctionnement en mode déconnecté

Les communications intra-NEMO utilisent des chemins complètement inclus dans le réseau NEMO. Elles sont donc indépendantes de la connectivité du réseau NEMO à l'Internet.

5.5.6 Mobilité globale dans l'Internet

Il est attendu que le déploiement du *multicast* IP se fasse à grande échelle dans la nouvelle génération d'Internet. L'utilisation du protocole MLD au niveau des routeurs mobiles pour adhérer au nom des MNNs aux sessions *multicast* fournit une mobilité globale dans l'Internet, puisque tous les routeurs IPv6 *multicast* sur les liens visités utilisent ce protocole de gestion de groupes.

5.5.7 Compatibilité avec le modèle ASM

Normalement, le protocole Xcast+6 ne permet de supporter que des sessions à source spécifique (SSM) car l'enregistrement se fait en envoyant une demande destinée à la source. Cependant, le modèle ASM peut être facilement supporté par notre solution. En effet, il suffit de configurer les DRs de destinations à l'intérieur du réseau mobile de façon à envoyer toutes les demandes d'enregistrement Xcast+6 correspondant à des adhésions ASM vers le MR. Ce dernier aura donc à maintenir une liste Xcast+6 de DRs de destinations par session ASM et acheminer le trafic *multicast* envoyé au groupe en question vers cette liste, quelque soit sa provenance.

5.6 Conclusion

Dans ce chapitre, nous avons proposé une solution permettant de fournir un service *multicast* aux réseaux NEMO dans IPv6. Cette solution utilise une transmission en Xcast+6 à l'intérieur du réseau mobile.

Nous avons commencé par définir une solution de base qui ne tient pas en compte la question d'emboîtement de la mobilité. Loin de vouloir établir une communication en *multi-unicast* explicite de bout en bout, nous avons défini un plan de contrôle où le routeur mobile effectue les transitions entre la transmission en *multicast* à l'extérieur du réseau NEMO et en Xcast+6 à l'intérieur. De plus, si le membre et la source sont tous les deux situés dans le même réseau NEMO, un chemin direct de livraison est établi. Dans le cas d'une source *multicast* située à l'intérieur du réseau NEMO, le trafic doit être d'abord tunnelé par le MR à son HA avant d'être transmis vers les membres externes via l'arbre de livraison *multicast*.

Nous avons ensuite étudié l'impact de l'emboîtement sur le comportement de notre solution de base. Ceci nous a permis de déceler un problème de non optimalité des chemins dans certains cas de communications intra-NEMO, du à l'absence d'une vision globale de la topologie dans un réseau NEMO emboîté. Pour remédier à ce problème, nous avons défini un algorithme d'exploration qui permet à un MR de localiser les sources *multicast* par rapport à sa propre arborescence. Cet algorithme utilise un nouveau type de message que nous avons appelé *Find Request*, transporté dans un en-tête d'extension de destination d'IPv6. Un MR déclenche une procédure d'exploration de son arborescence relative à une session *multicast* donnée en envoyant un paquet Xcast+6 d'interrogation contenant le message *Find Request* à tous ses sous-MRs directs. La procédure se poursuit de manière récursive de MR-père à MR-fils sur l'arborescence explorée. Cette procédure a pour résultat de déterminer l'emplacement relatif de la source par rapport au MR en question, et d'effectuer les enregistrements nécessaires s'il s'agit d'une source emboîtée à son arborescence.

Nous avons ensuite décris les traitements que doit effectuer un MR après chaque *handover* pour garder la continuité des sessions *multicast* des membres et sources situés derrière lui. Ces traitements sont de deux types : enregistrement à distance et notification des déplacements des sources.

Nous avons enfin entamé une évaluation théorique de notre solution en se basant sur différents critères, à savoir la nature de la livraison, la qualité des chemins, l'absence de boucles de transmission, la transparence vis-à-vis des MNNs, le fonctionnement en mode déconnecté, la mobilité globale dans l'Internet et la compatibilité avec le modèle ASM.

Conclusion

L'objectif de ce travail de recherche est de fournir un support *multicast* pour les réseaux mobiles dans IPv6. Les réseaux mobiles ou NEMO (*Network Mobility*) sont des réseaux connectés à l'Internet via des routeurs mobiles (MRs), capables de changer leur point d'ancrage à la topologie d'Internet.

Nous avons, en premier lieu, décrit l'état de l'art concernant le *multicast* IP, le mode de communication multipoint le plus ancien, basé sur l'utilisation d'un identificateur de groupe.

Nous avons ensuite enchaîné avec une présentation d'un nouveau mode de communication multipoint appelé le *multi-unicast* explicite, utilisant un encodage explicite de la liste des adresses destination dans les paquets de données. Nous avons décrit la spécification de base du routage *multi-unicast* explicite fournie par le protocole Xcast [BOI05], ainsi que les protocoles Xcast+ [MYU01] et GXcast [BOU03] qui sont des extensions du premier protocole.

Nous nous sommes ensuite intéressés à l'état de l'art concernant la mobilité des réseaux dans IPv6. Nous avons alors présenté la problématique des réseaux NEMO comparée à celle des nœuds, la terminologie associée, ainsi que le support de base de NEMO [RFC3963] récemment standardisé par l'IETF.

Nous avons, par la suite, entamé une étude de l'existant concernant le support du *multicast* dans les environnements NEMO. Ceci nous a permis en particulier de déceler la problématique du sujet comparé à celui du *multicast* pour nœuds mobiles, et de réaliser une critique argumentée de l'unique solution déjà proposée, basée entre autres sur le déploiement de *proxies* MLD à l'intérieur du réseau NEMO. Par cette critique, nous avons montré que cette solution présente des problèmes allant de la redondance de trafic et la construction inutile de branches *multicast* au risque de création de boucles de transmissions.

Nous avons ensuite proposé une nouvelle approche pour le support du *multicast* IP dans les réseaux NEMO. Notre solution se base sur une combinaison entre une transmission en *multi-unicast* explicite à l'intérieur du réseau NEMO et les approches classiques de gestion des nœuds *multicast* mobiles, à savoir la l'enregistrement à distance et le *tunnelling* bi-directionnel. Elle fournit un support du *multicast* pour les membres et les sources situés à l'intérieur d'un réseau mobile. Nous avons commencé par mettre en place une solution de base qui suppose l'absence d'emboîtement de la mobilité. Cette solution utilise le protocole Xcast+6, version de Xcast+ dédiée à IPv6 pour la livraison du trafic *multicast* à l'intérieur du réseau mobile. Nous avons ensuite

étudié l'impact de l'emboîtement de la mobilité sur le comportement de notre solution de base. Cette étude nous a permis de conclure qu'une non optimalité des chemins a lieu dans certains scénarios de communications intra-NEMO, à cause de l'absence d'une vision globale de la topologie dans un réseau mobile emboîté. Pour pallier à ce problème, nous avons mis en place un algorithme d'exploration permettant aux MRs situés dans un réseau mobile emboîté de localiser les sources *multicast* sollicitées, par rapport à leurs propres arborescences. Pour ce faire, nous avons été amenés à définir un nouveau type d'option pour l'en-tête d'extension de destination d'IPv6. Par la suite, nous avons défini les actions à effectuer après chaque *handover* du MR. Notamment, ces actions ont pour but de garder les sessions *multicast* en cours des nœuds situés dans le réseau NEMO. Nous avons enfin réalisé une évaluation théorique de notre solution en se basant sur différents critères.

Alors qu'un support de base pour NEMO a été déjà standardisé par l'IETF, les enjeux du *multicast* pour NEMO sont très peu invoqués par la littérature. Ce travail prend alors de l'importance du fait qu'il soit parmi les premiers travaux de recherches traitant du problème du *multicast* pour NEMO.

Nous avons réalisé une évaluation purement théorique de notre solution, en se basant sur des critères qualitatifs tels que l'absence de boucles de transmissions. Il serait intéressant d'effectuer ou des simulations pour mesurer certains paramètres, comme la latence d'adhésion aux groupes.

De plus, le protocole Xcast+6 étant facile à déployer car il se base sur les tables de routage *unicast*, notre solution pourrait être implémentée sur une plateforme réelle de test.

Par ailleurs, l'algorithme d'exploration que nous avons défini est basé sur l'envoi de paquets Xcast+6 d'interrogation de MR père à sous-MR. Ceci pourrait inspirer une idée similaire pour l'optimisation du routage unicast intra-NEMO au sein d'un réseau NEMO emboîté.

Bibliographie

[ADA03] A. Adams, J. Nicholas, and W. Siadak, "Protocol Independent Multicast — Dense Mode (PIM-DM) : Protocol Specification (Revised)", IETF Internet Draft, Septembre 2003, Work in progress.

[BET00] C. Bettstetter, A. Riedl, and G. Geßler, "Interoperation of Mobile IPv6 and Protocol Independent Multicast Dense Mode", Toronto, Canada, August 2000.

[BOI05] R. Boivie, N. Feldman, Y. Imai, W. Livens, D. Ooms, O. Paridaens, E. Muramoto. "Explicit Multicast (Xcast) Basic Specification", IETF Internet Draft, July 2005, Work in progress.

[BOU03] A.Boudani, A.Guitton, B.Cousin, "Generalized Explicit Multicast Routing Protocol", Draft, 2003, Work in progress.

[DEE91] S. Deering, "Multicast Routing in a datagram internetwork", PhD thesis, December 1991.

[DEE96] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu, L.Wei, "The Pim Architecture for Wide-area Multicast Routing", ACM Transactions on Networks, April 1996.

[ERN01] T.Ernst, "Le Support des Réseaux Mobiles dans IPv6", Université Joseph Fourier, PhD thesis, Octobre 2001.

[ERN03] T.Ernst, "Le Support des Réseaux Mobiles dans IPv6", CFIP : Colloque Francophone sur l'Ingenierie des Protocoles, Octobre 2003, Paris

[ERN05a] T.Ernst, H.-Y. LACH, "Network Mobility Support Terminology", IETF Internet Draft, Octobre 2005, IETF, Work in progress.

[ERN05b] T.Ernst, "Network Mobility Support Requirements", IETF Internet Draft, October 2005, Work in progress.

[FEN02] B. Fenner et al., "Protocol Independent Multicast — Sparse Mode (PIM-SM) : Protocol Specification (Revised)", IETF Internet Draft, Mars 2002, Work in progress.

[FEN04] B. Fenner, He. Haixiang, B. Haberman, H. Sandick, "IGMP/MLD-based Multicast Forwarding ("IGMP/MLD Proxying")", IETF Internet Draft, April 2004, Work in progress.

[Hee00] Hee-Sook Shin, Young-Joo Suh and Dong-Hee Kwon, "Multicast Routing Protocol by Multicast Agent in Mobile Networks", In Proceedings of International Conference on Parallel Processing, pp. 271-278, August 2000.

[HOL03] H. Holbrook, B. Cain. "Source-Specific Multicast for IP", IETF Internet Draft, 2003, Work in progress.

[JAN04] C. Janneteau, E. Riou, A. Petrescu, A. Olivereau, H.-Y. Lach, "IPv6 Multicast for Mobile Networks with MLD-Proxy", IETF Internet Draft, April 2004, Work in progress.

[JEL02a] C. Jelger, T. Noel, "Supporting Mobile SSM Sources for IPv6 (MSSMv6)," IETF Internet Draft, January 2002, Work in progress.

[JEL02b] C. Jelger, T. Noel, "Supporting Mobile SSM Sources", GLOBECOM '02, IEEE Global Commun. Conf., Taipei, Taiwan, Nov. 2002.

[JEL03a] C. Jelger, T. Noel, "An Analysis of Multicast Delivery with Mobile Receivers", 14th IEEE Int'l. Symp. Pers., Indoor and Mobile Radio Commun. (PIMRC'03), September 2003, Beijing, China.

[JEL03b] C. Jelger, T. Noel, "Performance Evaluation of Multicast Transmissions with Mobile Sources", 11th IEEE Int'l. Conf. Networks (ICON'03), September 2003, Sydney, Australia.

[LIN02] C.R. Lin and K.-M. Wang, "Scalable multicast protocol in IP-based mobile networks", Wirel. Netw., vol.8, pp.27–36, 2002.

[MYU01] S. MyungKI, K. YongJin, P. KiShik, and K. SangHa. "Explicit Multicast Extension (Xcast+) for Efficient Multicast Packet Delivery". ETRI Journal, 23(4), December 2001.

[NOE02] T. Noel and J. J. Pansiot, "A Multicast Architecture for Mobile Nodes," Int'l. J. Comp. and Info. Science, vol. 3 no. 2, 2002.

[SOL03] H. Soliman et al., "Hierarchical Mobile IPv6 Mobility Management (HMIPv6), IETF Internet Draft, June 2003, Work in progress.

[THA03] D. Thaler, "Border Gateway Multicast Protocol (BGMP) : Protocol Specification", IETF Internet Draft, 2003, Work in progress.

[TIM97] Tim G. Harrison , Carey L. Williamson , Wayne L. Mackrell , Richard B. Bunt, "Mobile multicast (MoM) protocol : multicast support for mobile hosts", Proceedings of the 3rd annual ACM/IEEE international conference on Mobile computing and networking,1997, Budapest, Hungary.

[XYL97] G. Xylomenos, and G. Polyzos, "IP Multicast for Mobile Hosts", IEEE Communications Magazine,vol. 35, no. 1, pp. 54-58, Jan. 1997.

[RFC1058] C. Hedrick, "Routing Information Protocol", June 1988.

[RFC1075] D. Waitzman, C. Partridge, S.E. Deering, "Distance Vector Multi-cast Routing Protocol", RFC 1075, Novembre 1988.

[RFC1247] J. Moy, "OSPF Version 2", RFC 1247, July 1991.

[RFC1584] J. Moy, “Multicast Extensions to OSPF”, RFC 1584, 1994.

[RFC1771] Y.Rekhter and T.Li, “A Border Gateway Protocol 4 (BGP-4)”, RFC 1771, 1995.

[RFC2189] A. Ballardie, “Core Based Trees (CBT version 2) Multicast Routing (Protocol Specification)”, RFC 2189, 1997.

[RFC2201] A. Ballardie. “Core Based Trees (CBT) Multicast Routing Architecture”, RFC 2201, 1997.

[RFC2236] W. Fenner, “Internet Group Management Protocol, version 2”, RFC 2236, 1997.

[RFC2283] T.Bates, R.Chandra, D.Katz and Y.Rekhter, “Multiprotocol Extensions for BGP-4”, RFC 2283, 1998.

[RFC2362] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, L. Wei, “Protocol Independent Multicast-Sparse Mode (PIM-SM) : Protocol Specification”, RFC 2362, 1998.

[RFC2460] S. DEERING, R. HINDEN, “Internet Protocol Version 6 (IPv6) Specification”, RFC 2460, Décembre 1998.

[RFC2461] T. Narten, E. Nordmark, W. Simpson, “Neighbor Discovery for IP Version 6 (IPv6)”, RFC 2461, Décembre 1998.

[RFC2473] A. Conta, S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, Décembre 1998.

[RFC2710] S. Deering, W. Fenner, and B. Haberman, “Multicast Listener Discovery (MLD) for IPv6”, RFC 2710, Octobre 1999.

[RFC3306] B. Haberman, D. Thaler, “Unicast-Prefix-Based IPv6 Multicast Addresses”, RFC 3306, 2002.

[RFC3344] C. Perkins, Ed., “IP Mobility Support for IPv4”, RFC 3344, 2002.

[RFC3376] B. Cain, S. Deering, A. Thyagarajan, “Internet Group Management Protocol, version 3”, RFC 3376, 2002.

[RFC3569] S. Bhattacharyya, “An Overview of Source-Specific Multicast (SSM)”. RFC 3569, 2003.

[RFC3618] D. Meyer, B. Fenner, “Multicast Source Discovery Protocol (MSDP)”, RFC 3618, Octobre 2003.

[RFC3775] D. Johnson, C. Perkins, J. Arkko, “Mobility Support in IPv6”, RFC 3775, Juin 2004.

[RFC3810] R. Vida and L. Costa, “Multicast Listener Discovery Version 2 (MLDv2) for IPv6”, RFC 3810, Juin 2004.

[RFC3956] P. Savola and B. Haberman, “Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address”, RFC 3956, November 2004.

[RFC3963] Devarapalli, R. Wakikawa, A. Petrescu, P. Thubert, “Network Mobility (NEMO) Basic Support Protocol”, RFC 3963, Janvier 2005.

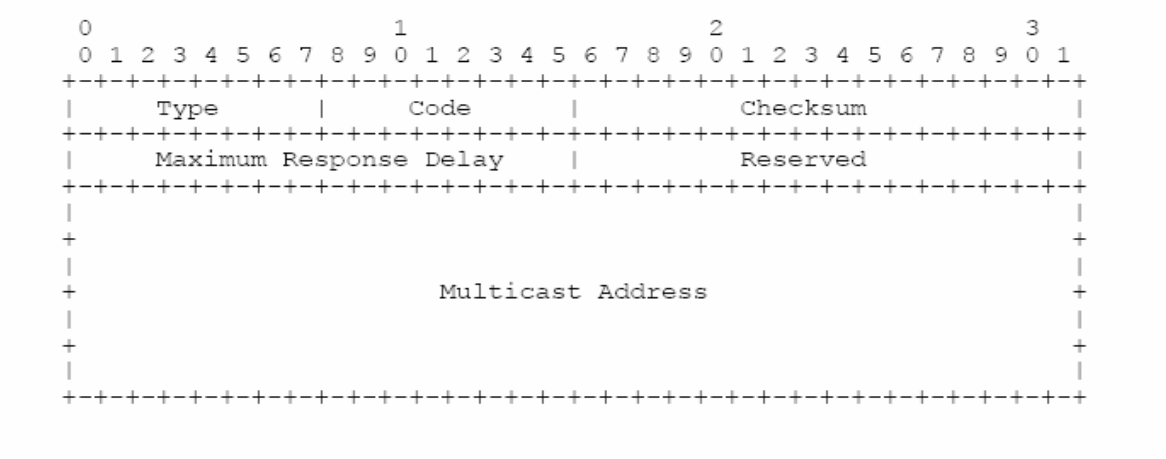
Annexe A

Format des Messages MLD

A.1 MLD version 1

Les messages MLD sont transportés dans des paquets ICMPv6 (champ *Next Header* du datagramme IP = 58).

Le format d'un paquet MLD est le suivant :



Format d'un paquet MLD

Voici la signification de chacun des champs.

Le champ type permet d'identifier trois types de messages MLD :

1. *Query Messages* (Type = 130 en décimal)
   (a) *General Query*
   (b) *Multicast-Address-Specific Query*
2. *Multicast Listener Report* (Type = 131 en décimal)
3. *Multicast Listener Done* (Type = 132 en décimal)

Le champ *Code* est initialisé à 0 par l'émetteur, et est ignoré par le destinataire.

Le champ *checksum* porte sur l'ensemble du message MLD ainsi que l'en-tête du message IPv6.

Le champ Maximum *Response Delay* permet de définir un délai de réponse maximum pour une réponse à un message *Query*. Pour les autres messages, il est initialisé à 0 par l'émetteur et ignoré par le destinataire.

Le champ reserved, n'est pas utilisé. Il est initialisé à 0 par l'émetteur et ignoré par le destinataire.

Le champ *Multicast Address* contient, suivant le type du message, une adresse IPv6 multicast ou est initialisé à 0 :

- pour un message de recensement général (*General Query*) ce champ est mis à zéro
- pour un message de recensement spécifique (*Multicast-Address-Specific Query*) il contient l'adresse multicast en question
- pour les messages de rapport (*Multicast Listener Report*) et de résiliation d'abonnement (*Multicast Listener Done*), le champ contient l'adresse multicast sur laquelle l'hôte souhaite écouter ou cesser d'écouter

A.2 MLD version 2

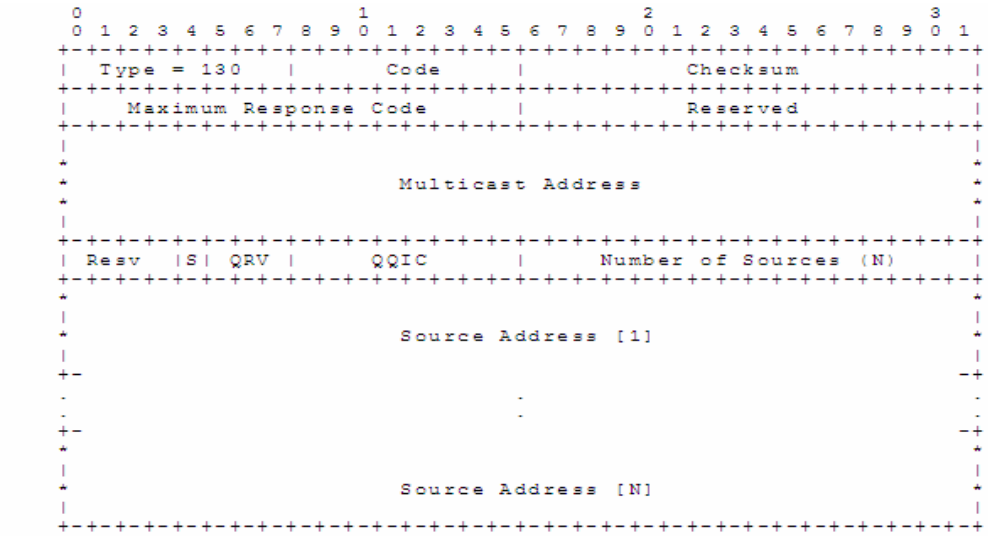
Il existe deux types de messages MLDv2 :

1. recensement des récepteurs multicast : type=130 (*Multicast Listener Query Message*)
2. rapport d'abonnement multicast version 2 : type=143 (

Pour garder l'interopérabilité avec la version précédente de MLD, les messages de rapport d'abonnement multicast version 1 et de résiliation d'abonnement multicast sont également supportés.

A.2.1 Messages de recensement MLDv2

Un message de recensement des récepteurs en MLDv2 est donné sur la figure Format d'un message de recensement MLDv2.



Format d'un message de recensement MLDv2

Les champs ont la signification suivante :
- type : le même type qu'en MLDv1
- code : mis à zéro par l'émetteur et ignoré par les récepteurs
- checksum : calculé de la même façon que pour la version précédente du protocole
- délai max. de réponse : utilisé pour calculer le délai maximal de réponse durant lequel le récepteur doit envoyer éventuellement son rapport d'abonnement
- inutilisé : mis à zéro par l'émetteur et ignoré par les récepteurs,
- adresse multicast sur laquelle porte le recensement,
- réservé : mis à zéro par l'émetteur et ignoré par les récepteurs,
- drapeau S : indique aux routeurs multicast qui reçoivent ce message s'ils doivent ou pas supprimer la mise à jour des temporisateurs, effectuée normalement au moment de la réception d'un message de recensement,
- QRV : contient la variable de robustesse utilisée par le recenseur (le nombre de fois qu'un récepteur envoie un rapport pour être robuste aux pertes dans le réseau),
- QQIC : code utilisé pour calculer l'intervalle de recensement,
- nombre de sources,
- adresse de la source [N], vecteur contenant la liste éventuelle des sources.

Trois types de messages de recensement de récepteurs multicast sont utilisés :

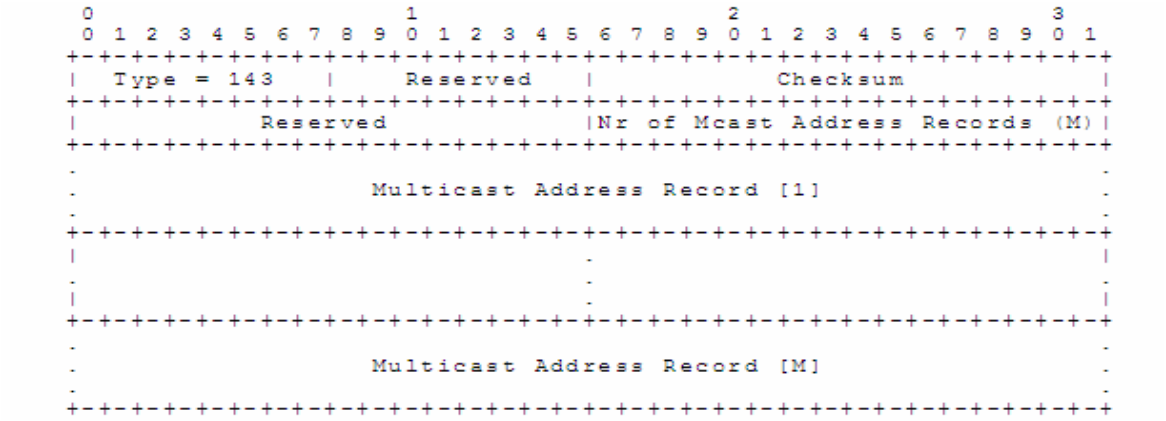
1. recensement général envoyé par un routeur multicast afin de découvrir les adresses multicast pour lesquelles il y a des récepteurs sur ses liens directs. Dans un tel message les champs "adresse multicast" et "nombre de sources" sont mis à zéro
2. recensement spécifique à une adresse multicast envoyé par un routeur multicast afin de découvrir l'existence de récepteurs pour une adresse multicast spécifique. Le champ "adresse multicast" contient l'adresse en question, tandis que le champ "nombre de sources" est mis à zéro

3. recensement spécifique à une adresse multicast et à une source envoyé par un routeur multicast afin de découvrir l'existence de récepteurs pour une adresse multicast et une source spécifiques. Le champ "adresse multicast" contient l'adresse en question, tandis que les champs "adresse source [i]" forment un vecteur de N adresses unicast (valeur spécifiée dans le champ "nombre de sources").

Les messages de recensement général sont envoyés à l'adresse de diffusion générale sur le lien (FF02 : :1). Les autres messages de recensement sont envoyés à l'adresse multicast spécifiée dans l'en-tête MLDv2.

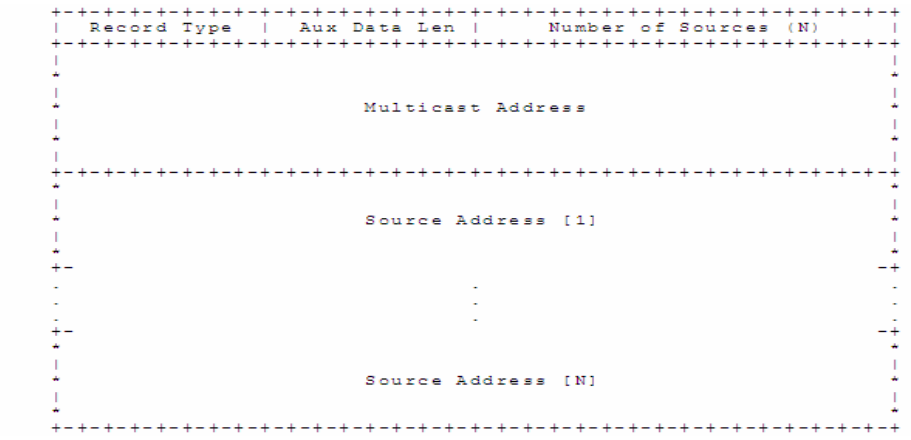
A.2.2 Rapports d'abonnement MLDv2

Un rapport d'abonnement multicast en MLDv2 est donné sur la figure Format d'un message de rapport d'abonnement MLDv2.



Format d'un rapport d'abonnement MLDv2

Les champs ont la signification suivante :
- type=143
- réservés, mis à zéro par l'émetteur et ignorés par les récepteurs
- checksum, calculé de la même façon que pour la version précédente du protocole
- nombre d'enregistrements d'adresse multicast
- enregistrement d'adresse multicast : chaque enregistrement d'adresse multicast a la forme donnée sur la figure suivante :



Enregistrement d'adresse multicast dans un rapport d'abonnement MLDv2

- Plusieurs types d'enregistrements d'adresse multicast peuvent être inclus dans un rapport d'abonnement :
  - Un "enregistrement d'état actuel" est envoyé par un hôte en réponse à un message de recensement. Il décrit l'état de l'hôte concernant une adresse multicast spécifique .Le champ "type d'enregistrement" peut dans ce cas avoir les valeurs *MODE_IS_INCLUDE* ou *MODE_IS_EXCLUDE*,
  - Un "enregistrement de changement de mode de filtrage" est envoyé par un hôte chaque fois qu'un appel EcouteIPv6Multicast modifie son mode de filtrage pour une adresse multicast précise. Le champ "type d'enregistrement" peut dans ce cas avoir les valeurs *CHANGE_TO_INCLUDE_MODE* ou *CHANGE_TO_EXCLUDE_MODE*,
  - Un "enregistrement de changement de la liste des sources" est envoyé par un hôte quand un appel EcouteIPv6Multicast modifie la liste des sources qu'il souhaite ou ne souhaite pas écouter pour une adresse multicast précise. Le champ "type d'enregistrement" peut dans ce cas avoir les valeurs *ALLOW_NEW_SOURCES* ou *BLOCK_OLD_SOURCES*.
- LDAux contient la longueur du champ données supplémentaires adresse multicast
- nombre de sources
- adresse source [i], vecteur contenant la liste des sources qui doivent être désormais autorisées ou bloquées
- données supplémentaires, si elles sont présentes, peuvent contenir des informations supplémentaires concernant l'enregistrement d'adresse multicast en question.

Annexe B

Encodage de l'en-tête Xcast6

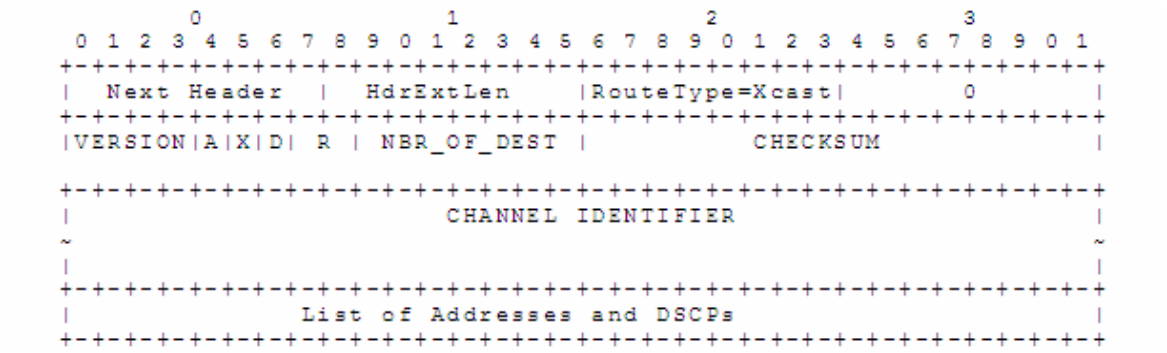
L'encodage de l'en-tête Xcast sur IPv6 (Xcast6) est semblable à celle sur IPv4, sauf qu'elle est contenue dans les en-têtes d'extension d'IPv6 :

[En-tête IPv6 | Xcast6 | en-tête de transport | charge utile]

- L'en-tête IPv6 doit porter dans le champ NextHeader la valeur "*Routing Extension*" (extension de routage). Le champ adresse source contient l'adresse de l'émetteur Xcast. Le champ adresse destination contient l'adresse *All_Xcast_Router*s.
- L'en-tête Xcast6 se compose également d'une partie fixe et de deux parties variables. La partie fixe et la première partie variable sont contenues dans une extension de routage. La deuxième partie variable est contenue dans une extension de destination.

B.1 L'en-tête extension de routage

Le bit P de Xcast4 n'est pas présent parce qu'il est implicite par la présence ou l'absence de l'extension de destination :



En-tête extension de routage d'un paquet Xcast+6

- *HdrExtLen* = la longueur de l'en-tête en 8 octets, ainsi un maximum de 127 destinations peut être énuméré (c'est pourquoi NBR_OF_DEST est de 7 bit).

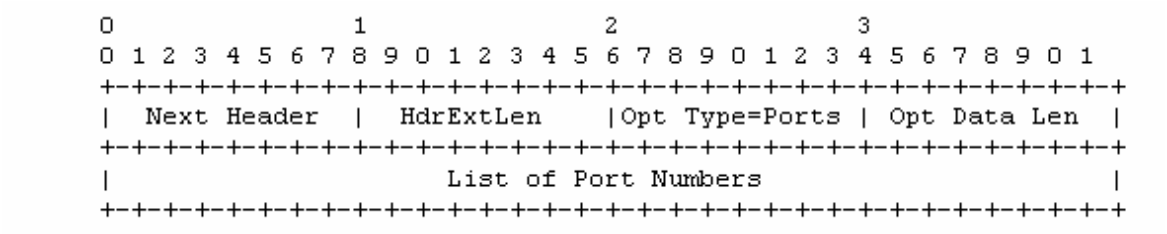
– *RouteType* = Xcast, devraient être assignés par IANA.
– Le quatrième octet est placé à 0.
– R = A réservé.
– Channel Identifier = identificateur du canal sur 16 octets.

Les autres champs sont similaires à ceux définis pour Xcast4.

La liste d'adresses et de DSCPs' est aligné sur 8-octet. La taille du bitmap est déterminée par le nombre de destinations et est un multiple de 64 bits.

B.2 L'en-tête extension de destinations

Cette partie est optionnelle. Elle contient la liste des ports. L'en-tête de destination n'est évalué que par le nœud destination.



En-tête extension de destinations d'un paquet Xcast+6

Le champ *Opt Type* (type d'option) devrait être assigné par IANA. Les trois premiers bits sont 010 pour indiquer que le paquet doit être jeté si l'option est inconnue et que l'option ne peut pas être changée en cours de route.

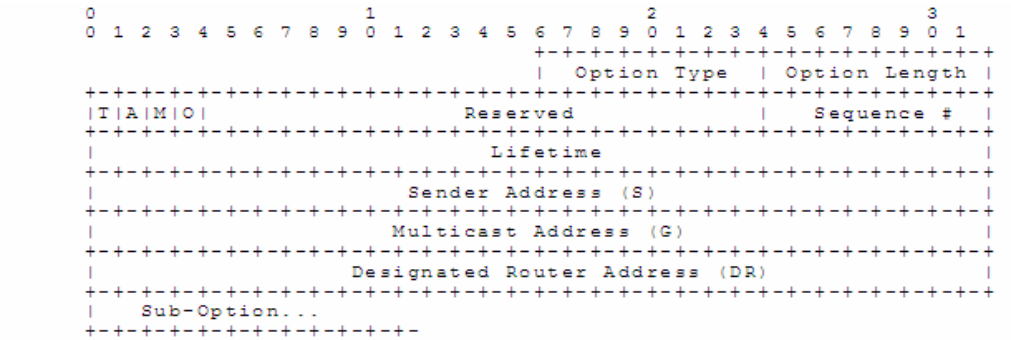
Le nombre de ports doit être égal au nombre de destinations indiqué dans l'en-tête de routage.

Annexe C

Messages de contrôle de Xcast+6

De nouvelles options "*Hop by Hop*" sont définies

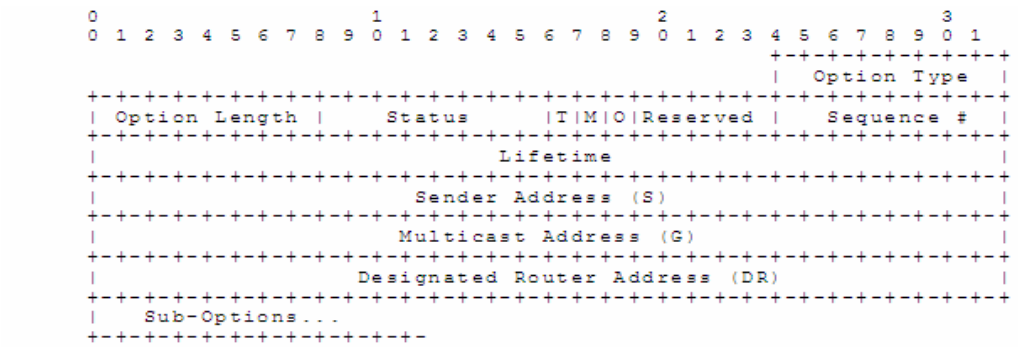
C.1 *Registration/De-registration Request*



Demande d'abonnement/désabonnement Xcast+6

- *Option Type* : TBD
- *Option Length* : entier non signé sur 8 bits
- T : type
  - 0 = *Registration Request*,
  - 1 = *Deregistration Request*
- A 1 = un acquittement doit être envoyé
- M 1 = *Registration Update* (X+MIP)
- O 1 = une sous Option IDO est inclue
- Sequence # nombre de 8 bite pour ordonnancer les messages
- ifetime entier non signé sur 32 bits

C.2 *Registration/Deregistration Reply*

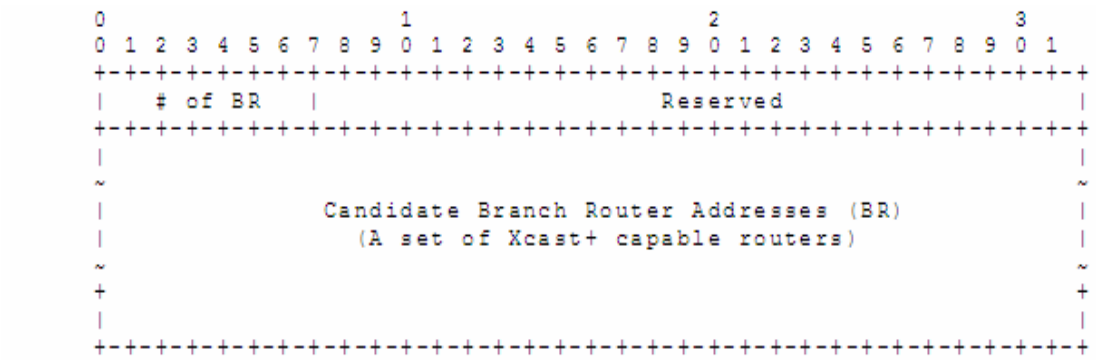


Confirmation d'abonnement/désabonnement Xcast+6

- *Option Type* TBD (To be attributed)
- *Option Length* entier non signé sur 8 bits
- Status entier non signé sur 8 bits indiquant l'état
  - 0 = *Registration Reply* accepté
  - 1 = *De-registration Reply* accepté
  - 128 = non accepté
- F :
  - 0 = *Registration Acknowledgement*
  - 1 = *Deregistration Acknowledgement*
- M 1 = *Registration Acknowledgement* de X+MIP
- O 1 = une sous Option IDO est inclue
- Sequence # nombre de 8 bits pour ordonnancer les messages
- Lifetime entier non signé sur 32 bits

C.3 Sous option IDO

Utilisée pour le déploiement progressif



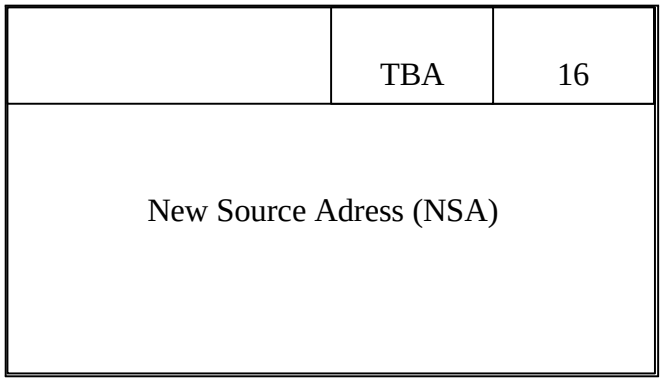
Sous option IDO d'un message de contrôle Xcast+

# of BR nombre d'adresses de routeurs de branchement candidats

Annexe D

Notification du handover d'une source SSM

Les auteurs dans [JEL02a] ont défini une nouvelle sous option de l'option IPv6 *binding update* appelée *SSM-Source Handover Notification*. Elle est utilisée pour notifier les récepteurs *multicast* du nouveau canal SSM associé à la source mobile après un *handover*. Le format de cette sous-option est le suivant :



Sous-option « SSM-Source Handover Notification »

Le champ "*new source address (NSA)*" contient la nouvelle adresse à utiliser par les récepteurs pour rejoindre le nouvel arbre *multicast* de livraison, associé au canal (NSA,G).

Le type de cette sous-option est à attribuer par l'IANA (TBA : *to be attributed*).